# MODELLE FÜR DIE SELBSTDIAGNOSE FEHLERTOLERANTER SYSTEME

Dissertation

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

der Fakultät für Physik
der Eberhard-Karls-Universität Tübingen

vorgelegt von

Eckhard Ammann

aus Reutlingen

1983

83 368

# MODELLE FÜR DIE SELBSTDIAGNOSE FEHLERTOLERANTER SYSTEME

Dissertation

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

der Fakultät für Physik
der Eberhard-Karls-Universität Tübingen

vorgelegt von

Eckhard Ammann

aus Reutlingen

1983

UNIV.
LEIDEN
BIBL.

| | |
|---|---|
| Tag der mündlichen Prüfung: | 6. Juni 1983 |
| Dekan: | Prof. Dr. A. Fäßler |
| 1. Berichterstatter: | Prof. Dr. M. Dal Cin |
| 2. Berichterstatter: | Prof. Dr. W. Güttinger |

# Abstract

AMMANN , ECKHARD :

MODELLE FÜR DIE SELBSTDIAGNOSE FEHLERTOLERANTER SYSTEME

Zur Beschreibung von Modellen für die Selbstdiagnose fehlertoleranter (Computer-) Systeme wird ein Raster entwickelt, das die beiden Teile der Selbstdiagnose, nämlich Test und Diagnose, nach ihren wesentlichen Merkmalen strukturiert.

Die sogenannten klassischen Modelle der Selbstdiagnose werden vollständig klassifiziert und untersucht. Dies schließt eine graphentheoretische Charakterisierung von t-diagnostizierbaren Diagnosestrukturen, die Angabe von Struktur- und Diagnosealgorithmen und von optimalen Diagnósestrukturen sowie Fragen der Realisierbarkeit des jeweiligen Modells ein.

Vergleichen von Ergebnissen ist eine natürliche Testphilosophie, die gut für eine funktionsbegleitende Überwachung von Systemteilen geeignet ist und in den klassischen Modellen noch nicht berücksichtigt wird. Modelle, die auf Vergleichstests aufbauen, werden im zweiten Teil der Arbeit eingeführt, untersucht und in Beziehung zu den klassischen Modellen gesetzt.

Die hier neu eingeführte Methode der dynamischen Diagnose verzichtet auf eine starre Diagnosestruktur, sondern organisiert den Testeinsatz in Abhängigkeit von der konkret vorliegenden Fehlersituation des Systems. Dies führt zu einer erheblichen Einsparung hinsichtlich der Zahl der benötigten Tests und der benötigten Testrunden. Verschiedene Algorithmen für die dynamische Diagnose werden vorgestellt, ihr Testaufwand im ungünstigsten Fall (worst case) und im Mittel (Erwartungswert) wird berechnet.

Als ein Konzept zur Realisierung und Validierung von Modellen der Selbstdiagnose wird das fehlertolerante Multi-Mikroprozessor-System ATTEMPTO vorgestellt, das in Tübingen entwickelt wird. ATTEMPTO ist als Arbeitsplatzrechner konzipiert und gibt dem Benutzer die Möglichkeit, für seine Probleme das angemessene Verhältnis von Systemdurchsatz und Fähigkeit zur Selbstdiagnose, die eine Voraussetzung für Fehlertoleranz ist, zu wählen.

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41547607_0007

INHALT :


| | Seite |
|---|---|
| 1. Einleitung | 1 |
| 2. Die Struktur von Selbstdiagnose-Modellen | 7 |
| 2.1. Beschreibung eines Tests | 8 |
| 2.2. Beschreibung der Diagnose | 11 |
| 2.3. Die Güte von Diagnosestrukturen | 16 |
| 2.4. Beispiele existierender Systeme und bekannter Modelle | 19 |
| 3. Die klassischen Modelle für die Selbstdiagnose | 24 |
| 3.1. Bezeichnungen und Reduktionen | 25 |
| 3.2. Untersuchung der klassischen Modelle bezüglich t-Diagnostizierbarkeit | 30 |
| 3.3. Untersuchung der klassischen Modelle bezüglich t-Detektierbarkeit | 45 |
| 3.4. Beziehungen zwischen den Modellen | 46 |
| 4. Statische Vergleichstestmodelle | 50 |
| 4.1. Bezeichnungen und Modellannahmen | 51 |
| 4.2. Das Vergleichstestmodell VT1 | 53 |
| 4.3. Das Vergleichstestmodell VT2 | 57 |
| 4.4. Das Vergleichstestmodell VT3 | 68 |
| 4.5. Anmerkungen zu den Vergleichstestmodellen | 69 |
| 5. Dynamische Diagnose | 72 |
| 5.1. Dynamische Diagnosealgorithmen | 73 |
| 5.2. Eine Gegenüberstellung von statischer und dynamischer Diagnose | 87 |
| 6. ATTEMPTO: ein Konzept zur Realisierung eines Selbstdiagnose-Modells | 91 |
| 6.1. Die Einordnung von ATTEMPTO in das Selbstdiagnose-Parametersystem | 92 |
| 6.2. Das Fehlertoleranzkonzept von ATTEMPTO | 93 |
| 6.3. Anmerkungen zu ATTEMPTO | 98 |
| 7. Ausblick | 99 |
| Anhang: PASCAL-Programme dreier dynamischer Diagnosealgorithmen | 100 |
| Referenzliste | 104 |

# 1. EINLEITUNG

Untersuchungsgegenstand dieser Arbeit ist die Selbstdiagnose fehlertoleranter (Computer-) Systeme, Systeme also, die auch bei Vorhandensein von Fehlern korrekt arbeiten und die Fähigkeit besitzen, selbst Fehler zu erkennen und zu lokalisieren.
Der Einsatz immer komplexerer Systeme (z.B. Multi-Mikroprozessor-Systeme), die immer weniger testbar sind und zugleich die Gefahr von Entwurfsfehlern bieten, anstehende Aufgaben mit sehr hohen Zuverlässigkeitsanforderungen (z.B. in der Raumfahrt oder in Intensivstationen) sowie stets auftretende flüchtige Fehler (z.B. durch α-Teilchenbefall) lassen eine Fehlervorbeugung etwa durch immer zuverlässigere Systemkomponenten (Fehlerintoleranz) nicht mehr ausreichend erscheinen, sondern verlangen nach Fehlertoleranzeigenschaften der Systeme.

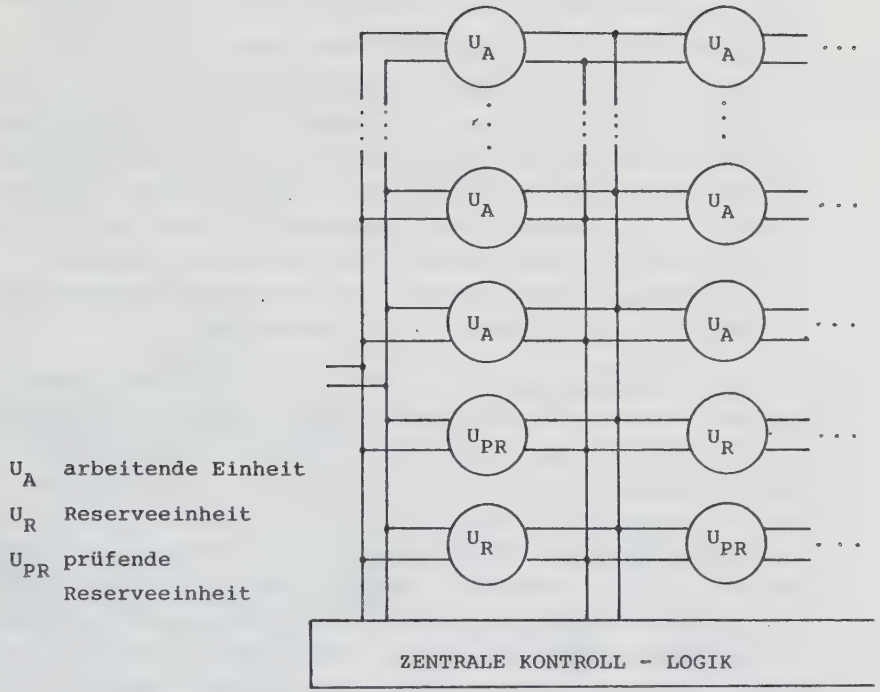
Dabei spricht man von statischer Fehlertoleranz [24], falls durch die Verteilung von Systemaufgaben auf redundante Komponenten und durch anschließende Mehrheitsentscheidung (Voten) Fehler maskiert werden. Dies hat den Vorteil einer funktionsbegleitenden Systemüberwachung und eines geringen Verwaltungsaufwands, bedeutet jedoch einen hohen Hardwareaufwand und schützt nicht gegen eine Akkumulation von Fehlern. Dynamische Fehlertoleranz [24] dagegen bedeutet aktives Vorgehen gegen Fehler, was im einzelnen das Testen zur Fehlererkennung, die Diagnose zur Fehlerlokalisierung, die Rekonfiguration und die Zustandsrestauration umfaßt. Dies bedeutet gegenüber dem statischen Fall ein besseres Ausnutzen der Resourcen, eine größere Flexibilität und die Vermeidung einer Akkumulation von Fehlern, aber als Nachteile auch einen wesentlich höheren Verwaltungsaufwand und eine starke Abhängigkeit von der Güte der benutzten Tests.

Die schwindende Beobachtbarkeit von Systemen von außen (Hochintegration) bringt die Notwendigkeit mit sich, Maßnahmen gegen Fehler nicht mehr von außen (etwa durch den Systembenutzer oder -betreiber) zu treffen, sondern sie in das System selbst einzubauen. Tests müssen deshalb bereits im System vorgesehen sein ("built-in-tests" [21]) und das System muß die Fehlerdiagnose

selbst durchführen können, also zur Selbstdiagnose fähig sein. Dies bringt auf der einen Seite zwar den Vorteil der "Automatisierung" der Fehlertoleranz, auf der anderen Seite aber auch die Gefahr, daß die zur Selbstdiagnose nötigen Systemteile selbst fehlerhaft sein können und so keine (korrekte) Fehlerbehandlung mehr möglich ist.

Das Treffen von Maßnahmen gegen Fehler setzt voraus, diesen Begriff zu definieren. Versteht man unter einem System eine Menge von Einheiten (Komponenten), die unter der Kontrolle eines Designs zusammenwirken [6], so ist ein Fehler eines Systems in dieser Arbeit als Abweichung des Systemverhaltens von einem festgelegten Sollverhalten definiert. Korrektes Verhalten bedeutet dabei nicht nur eine Ausgabe von korrekten Daten an die Systemumgebung, sondern auch stets korrekte Zustandsübergänge aller Systemteile. Der so festgelegte Fehlerbegriff deckt damit die in der amerikanischen Literatur gebräuchlichen Begriffe "failure-fault-error" [78] ab.

Als konkretes Beispiel eines selbstdiagnostizierenden, fehlertoleranten Systems sei MICRONET beschrieben (nach [72]): MICRONET ist ein parallel arbeitender Kommunikationsrechner, der in der Lage ist, durch Parallelität und Pipelining eine Vielzahl anfallender Daten on-line zu bearbeiten. Gruppen von Computern, die gleichartige Berechnungen parallel durchführen, bilden Strings. Alle Computer eines Strings empfangen Daten über einen gemeinsamen (redundanten) Inputbus und wickeln ihren Output über einen gemeinsamen (redundanten) Outputbus ab. In jedem String gibt es Reservecomputer, verschiedene Strings werden seriell hintereinandergeschaltet (Figur 1.1.). Einer der Reservecomputer in jedem String fungiert als Prüfeinheit und testet nach und nach alle arbeitenden Computer seines Strings, indem er ebenfalls die gerade anliegenden Daten berechnet und seine Ergebnisse mit denen des arbeitenden Computers vergleicht. Falls genau eine arbeitende Einheit von der Prüfeinheit abweicht, gilt sie als fehlerhaft und wird durch eine Reserveeinheit ersetzt. Existieren Abweichungen zu mehreren aktiven Einheiten, so wird die Prüfeinheit selbst als fehlerhaft betrachtet und abgeschaltet; eine andere Reserveeinheit übernimmt ihre Funktion. Diese Rekonfiguration wird durch eine zentrale Kontroll-Logik vorgenommen. Die durch den




Figur 1.1. Die Struktur von MICRONET

Ausfall einer aktiven Einheit verloren gegangenen Daten können nicht wiedererlangt werden, doch kann das Ausmaß dieses Verlustes als erträglich angesehen werden.

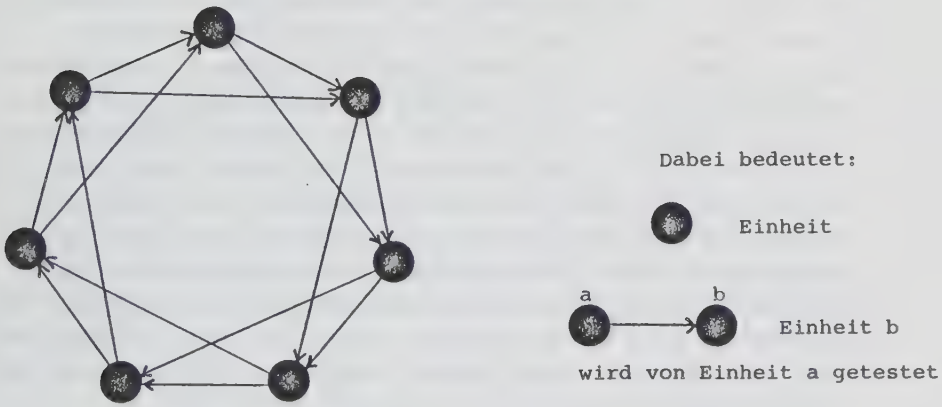
Wie können nun die Selbstdiagnoseeigenschaften von MICRONET, allgemeiner von irgendeinem System, beschrieben werden? Grundsätzliche Aspekte der Fehlertoleranz wurden in [10] und [70] diskutiert, [37] und [17] beschränken sich auf Software- bzw. Hardware-Fehlertoleranz. Angeregt durch die Arbeit von Preparata, Metze und Chien [69] setzte eine rege Forschungstätigkeit zur Beschreibung von Modellen der (Selbst-) Diagnose und ihrer

Eigenschaften ein. Ein Überblick über diese theoretischen Modelle wird in [32,56] gegeben. Neue Ansätze, seien sie theoretischer oder praktischer Natur, vornehmlich aus dem deutschsprachigen Raum sind in [25] zu finden.

Vier Ziele lassen sich für meine Arbeit angeben:
Zunächst soll ein allgemeiner Rahmen für Selbstdiagnosemodelle erarbeitet werden. Daraufhin werden eine Reihe neuer Modelle für die Selbstdiagnose vorgestellt und diskutiert, die alle eine starre Diagnosestruktur verwenden. Drittes Ziel ist die Einführung der neuen Methode einer dynamischen Diagnose. Abschließend wird ein Konzept zur Realisierung eines der vorgestellten theoretischen Modelle präsentiert.

In Kapitel 2 werden wesentliche Merkmale von Test und (Selbst-) Diagnose extrahiert. Prüfobjekte von Tests können dabei Einheiten wie vollständige Computer, Prozessoren, Hintergrundspeicher, Kommunikationsverbindungen oder auch Mengen dieser Einheiten sein. Die Gesamtheit dieser Merkmale bildet dann eine Art "Parametersystem" der Selbstdiagnose. Auf der einen Seite können damit vorliegende Systeme und Modelle genau beschrieben und vor allem auch miteinander verglichen werden. Von großer Bedeutung ist auf der anderen Seite auch die Unterstützung, die dieses Parametersystem beim Entwurf neuer Modelle oder Systeme bieten kann.
Wesentliche Merkmale von Tests sind nicht nur ihre Beschaffenheit (wer testet wen wie ?) und die Organisation des Testeinsatzes, sondern auch die Annahmen, die der Test über die Beschaffenheit der betroffenen Testobjekte (Einheiten) und über mögliche (von ihm erfaßte) Fehler in den Einheiten macht.
Die anschließende Diagnose muß zum einen die Familie der durchzuführenden Tests festlegen und zum zweiten nach Durchführung der Tests deren Ergebnisse auswerten. Verwendet ein Modell eine feste Diagnosestruktur, so lassen sich Maße für die Güte dieser Struktur zu Diagnosezwecken angeben. Diese Maße sind geeignet, verschiedene Diagnosestrukturen, aber auch verschiedene Modelle anhand einer gegebenen Diagnosestruktur, miteinander zu vergleichen. Den Abschluß des Kapitels bilden kurze Beschreibungen einiger bekannter Systeme und Modelle und ihre Einfügung in das vorgeschlagene Parametersystem.

Die sogenannten "klassischen" Modelle der Selbstdiagnose zeichnen sich im wesentlichen durch die Verwendung von Tests aus, die genau zwei Einheiten betreffen (eine testende und eine getestete Einheit). Die Bezeichnung "klassisch" rührt dabei von der prinzipiellen Ähnlichkeit dieser Modelle mit dem ersten bekannten Modell, dem von Preparata et al., her. Diagnosestrukturen dieser Modelle lassen sich als gerichtete Graphen, sogenannte Diagnosegraphen, darstellen; einen typischen Diagnosegraphen zeigt Figur 1.2..




Figur 1.2. ein Diagnosegraph

Kapitel 3 klassifiziert und untersucht alle möglichen klassischen Modelle, unter denen sich auch das schon erwähnte Modell von Preparata et al. befindet. Verwendete Maße sind dabei die Diagnostizierbarkeit (zur Fehlerlokalisierung) und die Detektierbarkeit (zur bloßen Fehlererkennung) von Diagnosestrukturen; diese Maße werden graphentheoretisch interpretiert. Ein Vergleich der verschiedenen Modelle untereinander ergibt Verbandsstrukturen auf der Menge der Modelle.

Konventionelle Tests verlangen nach speziellen Testprogrammen, die außerhalb normaler Arbeitsphasen durchlaufen werden. Eine

andere Testphilosophie , die auf einem Vergleichen von Einheiten bzw. deren Ergebnissen beruht, stellt Kapitel 4 vor. Da prinzipiell überhaupt jeder Test aus einem Vergleich eines Verhaltens mit einem vorgegebenen Sollverhalten besteht, ist eine Modellierung dieses Sachverhalts als Vergleichstest nur natürlich. Wesentlicher Vorteil von Vergleichstests gegenüber konventionellen Tests ist die Möglichkeit eines funktionsbegleitenden Einsatzes, d.h. die Möglichkeit, die bei der normalen Auftragsbearbeitung anfallenden Daten zugleich auch als Testdaten zu verwenden.

Während die in den Kapiteln 3 und 4 eingeführten und untersuchten Modelle stets feste Diagnosestrukturen vorsehen, verzichtet die in Kapitel 5 vorgestellte neue Methode der dynamischen Diagnose auf diese Festlegung. Test- und Diagnosephasen wechseln sich solange ab, bis die Diagnose vollständig ist. Das Ergebnis jedes einzelnen Tests wird in der anschließenden Diagnosephase sofort ausgewertet, die Festlegung des nächsten Tests erfolgt in Abhängigkeit der schon vorliegenden Diagnoseinformation. Auf diese Weise baut sich nach und nach eine vom jeweiligen Fehlerzustand des Systems abhängige spezielle Diagnosestruktur auf. Entscheidender Vorteil dieses dynamischen Diagnose gegenüber der herkömmlichen (statischen) Diagnose ist, daß durch die Anpassung des Testeinsatzes an die konkret vorliegende Fehlersituation ein wesentlicher Anteil an Tests eingespart werden kann.

Kapitel 6 schließlich stellt ATTEMPTO vor, ein unter meiner Mitwirkung in Entwicklung befindliches fehlertolerantes Multi-Mikroprozessor-System. ATTEMPTO stellt einen Versuch dar, ein Modell für die Selbstdiagnose (in diesem Fall das in Abschnitt 4.3. beschriebene Modell VT2) in einem konkreten System zu realisieren. Zum einen sollen dabei neue Anhaltspunkte für eine weitere (realitätsnahe) Modellentwicklung gewonnen werden, zum anderen der Nachweis der Einsetzbarkeit eines theoretischen Modells der Selbstdiagnose erbracht werden.

Bezeichnungen:

| | |
|---|---|
| $\|N\|$ | Kardinalität der Menge N |
| $\lceil x \rceil$ | kleinste ganze Zahl $\geq x$ |
| $\lfloor x \rfloor$ | größte ganze Zahl $\leq x$ |
| log | Logarithmus zur Basis 2 |

# 2. DIE STRUKTUR VON SELBSTDIAGNOSEMODELLEN

Ein Modell für die Selbstdiagnose fehlertoleranter Systeme zerfällt notwendigerweise in die beiden Teile Testmodell und Diagnosemodell. Zuerst muß geklärt sein, wie und unter welchen Annahmen die zur Fehlerdiagnose nötigen Signale bzw. Daten grundsätzlich gewonnen werden (Testmodell). Dabei muß aber nicht nur über die Beschaffenheit der Tests selbst und die Art ihres Einsatzes eine Aussage gemacht werden, sondern auch über die Objekte (Einheiten), die durch sie getestet werden, und über die betrachteten möglichen Fehler (Abschnitt 2.1.).
Das Modell der Diagnose beschreibt dann, wie derart gestaltete Tests eingesetzt werden und wie die durch sie gewonnene Information ausgewertet wird (Abschnitt 2.2.). Die Struktur eines Selbstdiagnosemodells ist dann durch die Wahl der in 2.1. und 2.2. beschriebenen "Parameter" festgelegt.
Abschnitt 2.3. führt bekannte und mögliche neue Maße für die Güte der in der Diagnosephase verwendeten Diagnosestrukturen an. Schließlich werden in Abschnitt 2.4. einige bekannte Systeme und Modelle in das "Parametersystem" der Selbstdiagnose eingefügt.

## 2.1. BESCHREIBUNG EINES TESTS

Ein Test ist eine Vorkehrung, um an gewissen Objekten ein vom Sollverhalten abweichendes Verhalten festzustellen. Selbstdiagnose verlangt nach Tests, die nicht von einer Instanz außerhalb des Systems abhängen dürfen. Aus diesem Grunde wird der Einsatz von sogenannten "built-in-tests" [21] notwendig.

Das folgende Parametersystem beschreibt jeweils eine Art von Test. Sind in einem Modell der Selbstdiagnose verschiedene Arten von Tests vorgesehen, so muß für jede dieser Arten die folgende Beschreibung gegeben werden.
Hier zunächst die Parameter, die nachfolgenden Ausführungen erläutern die einzelnen Punkte näher:

2.1.1. Beschaffenheit der Einheiten, die mit dem Test in Verbindung stehen
- Art der Einheiten
- Fehler in den Einheiten sind stochastisch abhängig oder unabhängig
- verschiedene Einheiten fallen gleichwahrscheinlich aus oder nicht

2.1.2. Art der zugelassenen Fehler
- permanent , transient oder beide Arten von Fehlern

2.1.3. Beschaffenheit des Tests
- am Test beteiligt sind eine s-Menge A und eine r-Menge B von Einheiten, A und B sind disjunkt oder nicht.
- m-wertiger Test
- gerichteter oder ungerichteter Test
- vollständiger oder unvollständiger Test.

2.1.4. Organisation des Testeinsatzes
- der Testzeitpunkt ist funktionsbegleitend, periodisch oder sporadisch.

2.1.1.: Vom Test betroffene Einheiten in einem Computersystem können Prozessoren, Peripherieteile, Kommunikationsverbindungen (Links, Busse) oder auch Speicher sein. Auch komplexere Objekte wie Single-Board-Computer können als Einheiten angesehen werden. In einem homogenen Multiprozessor-System, in dem Prozessoren als Einheiten betrachtet werden, sind Fehler in verschiedenen

Einheiten in guter Näherung stochastisch unabhängig und gleichwahrscheinlich. Werden jedoch ein Prozessor und sein lokaler Speicher jeweils als Einheiten für einen Test angesehen, sind Fehler dieser beiden Einheiten sicher nicht mehr unabhängig. In inhomogenen Systemen sind Fehler in verschiedenen Einheiten im allgemeinen nicht gleichwahrscheinlich.

2.1.2.: Ein permanenter Fehler ist ein dauerhafter, stabiler Fehler. Im Unterschied dazu wird ein transienter Fehler durch einen zeitweisen Einfluß etwa von äußeren Umständen bewirkt und ist nach Abklingen dieser Umstände nicht mehr festzustellen. In der Literatur (etwa [78])wird diese Einteilung von Fehlern durch die Betrachtung von intermittierenden Fehlern noch weiter verfeinert. Intermittierende Fehler sind dann solche, die durch instabile Hardware oder durch seltene Hardware-Software-Konstellationen ausgelöst werden. Sind diese seltenen Konstellationen durch einen Test erfaßbar, kann man diese intermittierende Fehler den permanenten zuordnen, im anderen Fall den transienten. Modelle, die zur Erkennung auch von nicht permanenten Fehlern angelegt sind, werden in [57,58] vorgeschlagen.

2.1.3.: Jeder Test bewirkt eine Einteilung der Menge der beteiligten Einheiten in zwei Mengen A und B der Kardinalitäten s und r. In vielen Fällen kann dabei A als Menge der (aktiven) Tester und B als Menge der (passiven) getesteten Einheiten dieses Tests interpretiert werden. Bei einem "klassischen" Test etwa wird ein Einheit i von einer anderen Einheit j getestet. Die Parameter sind deshalb hier A={j}, s=1, B={i} und r=1, A und B sind disjunkt. Dagegen lauten die Angaben für einen Selbsttest eines Prozessors i A={i}=B, s=r=1 und nicht disjunkt.
Der Wert m gibt die Zahl der verschiedenen Zustände der betroffenen Einheiten an, die durch den Test unterschieden werden. Gilt m=2 wie in den klassischen Modellen, so wird nur zwischen intakter und defekter Einheit unterschieden. Besteht A aus der Menge der Tester und B aus der Menge der getesteten Einheiten bezüglich eines Tests, so heißt dieser Test gerichtet. Können hingegen die Funktionen der Mengen A und B nicht unterschieden werden, so ist der Test ungerichtet. Klassische Tests (z.B. in [69,11]) sind gerichtet. Besteht ein Test aus einem Vergleich zweier Einheiten, so ist er ungerichtet. Ein gerichteter Test heißt

vollständig, falls unter der Voraussetzung, daß A aus intakten Einheiten besteht, jeder Fehler in der Menge B durch den Test entdeckt wird, sonst unvollständig. Für einen ungerichteten Test bedeutet Vollständigkeit, daß der Test einen Fehler meldet unter der Voraussetzung, daß genau eine der beiden Mengen A und B defekte Einheiten enthält.
Einen Ansatz zur Einbeziehung von unvollständigen Tests stellt die Definition der sogenannten "Coverage" [79] dar. Darunter wird die Wahrscheinlichkeit verstanden, daß ein Fehler in der getesteten Einheit durch den Test entdeckt wird. Vollständige Tests haben also den Coverage-Wert 1.

2.1.4.: Ein periodischer Testeinsatz ist wohl der am häufigsten angewendete. Zu bestimmten Zeitpunkten wird die normale Systemtätigkeit unterbrochen und eine Testphase eingelegt. Sporadischer Testeinsatz erscheint dann sinnvoll, wenn der begründete Verdacht einer Systemfehlfunktion besteht. Ein Test ist funktionsbegleitend, wenn er die notwendigen Informationen aus den Ergebnissen bezieht, die etwa Prozessoren bei der Abarbeitung von Benutzerprogrammen bereitstellen. Testen kostet hier also keine zusätzliche Zeit. In der Mitte zwischen funktionsbegleitendem und periodischem Testeinsatz steht die für die "Roving-Diagnose" [1] benötigte Testorganisation: In einer Art Rotationsprinzip werden die Einheiten in aktive und zu testende Einheiten eingeteilt, so daß nach einer gewissen Zeit alle Einheiten des Systems getestet sind. Auf einer ähnlichen Idee beruhen die Vorschläge in [77] und [80].

Anmerkungen:

1. Die sehr einschneidenden Anforderungen an einen vollständigen Test lassen sich am ehesten verwirklichen, wenn ein funktionsbegleitender Testeinsatz gewählt wird und unter einem Fehler einer Einheit ihr Fehlverhalten während der Abarbeitung eines Benutzerprogramms verstanden wird.

2. Der Einsatz von fehlerkorrigierenden bzw. fehlererkennenden Codes [49], insbesondere gegen Speicherfehler, leidet am großen Aufwand für eine Codierung und Decodierung von Daten und an der Unverträglichkeit der Codes mit der stets implementierten Paritätskontrolle [66]. Codes, die diese Nachteile nicht mehr aufweisen und deshalb für fehlertolerante Computersysteme

geeignet sind, werden in [67] vorgestellt. Auf der Gatterebene können mit Hilfe von fehlererkennenden Codes "self-checking-circuits" [83], d.h. Schaltkreise, bei denen ein vorhandener Fehler zu einem verfälschten Output nach außen führt, konstruiert werden.
Bei jeder dieser Einsatzmöglichkeiten kann von Tests im Sinne dieses Abschnitts gesprochen werden.

3. In fast allen bekannten Modellen für die Selbstdiagnose werden als Einheiten in 2.1.1. nur Prozessoren, Speicher oder aus diesen Bausteinen gebildete komplexere Gebilde verstanden. Das in [45] vorgestellte Modell läßt in der Menge B auch eine Kommunikationsverbindung zu unter der Voraussetzung, daß die beiden Endknoten dieser Verbindung nicht in B liegen.

4. Modelle für Tests, in denen die Mengen A und B aus 2.1.3. nicht beide einelementige Mengen sind, werden in [73,74,75,38] vorgestellt. Ihnen gemeinsam ist, daß die Menge A der Tester aus mehreren Einheiten besteht (zur Durchführung eines Tests sind mehrere Betriebsmittel notwendig), während B nur aus einer getesteten Einheit besteht. So gesehen, stellen diese Tests Verallgemeinerungen der klassischen Tests dar.

## 2.2. BESCHREIBUNG DER DIAGNOSE

Diagnose bedeutet, die Menge der durchzuführenden Tests festzulegen und mit Hilfe der Ergebnisse der Tests jeder betroffenen Einheit im System einen Zustand zuzuweisen (bei Verwendung von 2-wertigen Tests etwa die Zustände intakt oder defekt).
Hier die Parameter der Diagnose:

### 2.2.1. Voraussetzungen der Diagnose

- t-Fehlerannahme oder probabilistische Methode
- intakte Einheit dürfen (dürfen nicht) als defekt diagnostiziert werden
- 1-Schritt-Diagnose oder Mehrschrittdiagnose
- Annahmen über die Testergebnisse

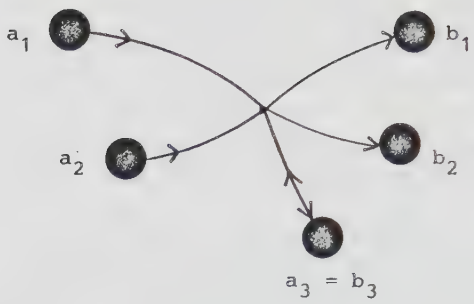
2.2.2. Festlegung der Testmenge
- statisch oder dynamisch
- Mehrfachanwendung eines Tests ist zulässig oder nicht

2.2.3. Organisation der Diagnose (d.h. der Auswertung der Testergebnisse)
- zentral oder dezentral
- globale oder lokale Sicht
- hierarchisch oder nicht hierarchisch

2.2.1.: Unter den Voraussetzungen der Diagnose sind alle Annahmen anzugeben, auf die sich die Diagnose bei der Auswertung der Testergebnisse stützt. Die Mehrzahl der bekannten Modelle (etwa [69,11,74,5]) machen die t-Fehlerannahme, d.h. diagnostizieren unter der Voraussetzung, daß höchstens t defekte Einheiten vorliegen. Wird diese Schranke t angemessen gewählt und ist bei statistischer Unabhängigkeit die Ausfallswahrscheinlichkeit kleiner als 0.5, so kann dann eine eindeutige Diagnose erreicht werden. Dagegen strebt die probabilistische Methode [55,16] eine Zustandszuweisung der betroffenen Einheiten an, die unter Berücksichtigung der Ausfallswahrscheinlichkeiten der Einheiten die höchste Wahrscheinlichkeit unter allen nach den Testergebnissen möglichen Zustandszuweisungen hat.
Im allgemeinen fordern Diagnosemodelle, daß intakte Einheiten nicht als defekt diagnostiziert werden dürfen. In [31] wird dagegen ein Modell vorgestellt, das bei Annahme von höchstens t Fehlern die Ersetzung von s≥t Einheiten gestattet. Eine gewisse Anzahl von Einheiten, deren Zustand nicht eindeutig bestimmt werden kann, kann also als defekt erklärt werden. Ebenfalls wird dies im Modell von Smith [81] zugelassen.
Die Unterscheidung zwischen 1-Schritt- und Mehrschrittdiagnose stammt bereits von [69] her. Dabei bedeutet 1-Schrittdiagnose, daß die Auswertung der Testergebnisse zur vollständigen Diagnose, d.h. zur Zustandszuweisung an alle betroffenen Einheiten, führt. Bei der Mehrschrittdiagnose gelingt im allgemeinen nur eine Teildiagnose. Die dabei als defekt erkannten Einheiten werden repariert oder ersetzt und danach eine neue Test- und Diagnoserunde begonnen. Dies wiederholt sich so oft, bis das gesamte System diagnostiziert ist.
Schließlich müssen noch Annahmen über die Testergebnisse gemacht

werden. Werden nur vollständige Tests verwendet (2.1.3.), so muß lediglich noch eine Annahme über die Testinvalidierung, d.h. über Testverfälschungen bei fehlerhaften Einheiten in der Menge A bzw. bei ungerichteten Tests in beiden Mengen A und B gemacht werden. Bei unvollständigen Tests müssen Angaben über zu erwartende Testergebnisse geliefert werden. Beispiele von Testannahmen in bekannten Modellen erscheinen in Abschnitt 2.4., weitere Beispiele zum Teil neuer Modelle in den Kapiteln 3 und 4.

2.2.2.: Alle bisher bekannten Modelle sehen eine statische Diagnose vor. Dabei wird, unabhängig vom konkreten Zustand des Systems, zu Beginn der Diagnose die Menge der durchzuführenden Tests festgelegt. Dagegen wird in einer dynamischen Diagnose nach jedem einzelnen durchgeführten Test die schon gewonnene Information ausgewertet und in Abhängigkeit davon, also auch in Abhängigkeit des konkreten Systemzustands, der nächste durchzuführende Test festgelegt, bis die Diagnose vollständig ist. Die Diagnosestruktur, die sich im dynamischen Fall im Laufe der Diagnose erst aufbaut und im statischen Fall bereits zu Beginn festliegt, kann dabei jeweils durch einen "ungerichteten" (für ungerichtete Tests) bzw. einen "gerichteten" ( für gerichtete Tests) "Hypergraphen" beschrieben werden. Nach Berge [14] sind (ungerichtete) Hypergraphen Graphen, deren Kanten nicht notwendigerweise zwei Endknoten haben. Ein gerichteter Hypergraph entsteht dann, wenn die inzidenten Knoten aller Kanten als Anfangs- oder Endknoten (oder beides) markiert werden.
Der gerichtete Hypergraph einer Diagnosestruktur wird nun gebildet, in dem die betroffenen Einheiten des Systems als Knoten angesehen werden und jeder vorgesehene Test als Kante die von ihm betroffenen Einheiten verbindet, wobei die Einheiten der Menge A als Anfangsknoten und die Einheiten der Menge B als Endknoten markiert werden. Zur Veranschaulichung sei der entsprechende Ausschnitt aus einer Diagnosestruktur für einen gerichteten Test mit $A=\{a_1,a_2,a_3\}$, $B=\{b_1,b_2,b_3\}$ mit $a_3=b_3$ (also A und B nicht disjunkt) in Figur 2.1. gezeigt.
Die Verwendung von klassischen Tests führt zu gerichteten gewöhnlichen Graphen, Selbsttests dagegen ergeben Schlingen in der Diagnosestruktur.
Eine weitere Vorgabe bei der Festlegung der Testmenge bedeutet die Angabe, ob ein Test mehrfach angewendet werden darf. Eine




Figur 2.1. Eine (verallgemeinerte) Kante in einem gerichteten Hypergraph.

Mehrfachanwendung von Tests ist besonders dann wirkungsvoll, wenn das Auftreten von transienten Fehlern zu erwarten ist. Ist eine solche Mehrfachanwendung ausgeschlossen, so stellt die Diagnosestruktur einen einfachen Hypergraphen [14,S.389], also einen Hypergraphen mit lauter verschiedenen Kanten dar.

2.2.3.: Von großer Bedeutung ist die Festlegung der Organisation der Diagnose. Hier die Definitionen zu den Parametern von 2.2.3.:
Die Diagnose ist zentral, falls eine einzige Instanz im System für die gesamte Durchführung der Diagnose verantwortlich ist, und dezentral , falls mehrere Instanzen an der Diagnose mitwirken. Von globaler Sicht sei gesprochen, falls jede an der Diagnose mitwirkende Instanz alle betroffenen Einheiten des Systems diagnostiziert, von lokaler Sicht, falls jede Diagnoseinstanz nur eine begrenzte Systemsicht hat. Eine Diagnosestrategie heißt hierarchisch, falls eines der beiden folgenden Merkmale a) oder b) zutrifft:

a) Im ersten Diagnoseschritt werden komplexe Einheiten diagnostiziert. Die als defekt erkannten Einheiten werden jetzt als Systeme kleinerer Einheiten angesehen und (genauer) diagnostiziert. Diese Verfeinerung der Diagnose erfolgt so

oft, wie sie vom Modell vorgegeben ist.

b) In einem ersten Diagnoseschritt werden zentrale Teile des Systems diagnostiziert und gegebenenfalls repariert oder ersetzt. Aufbauend darauf werden in den folgenden Diagnoseschritten nach und nach die weniger zentralen Teile des Systems diagnostiziert

Dabei bleibt natürlich ungeklärt, durch welche Instanz die zentralen Teile diagnostiziert werden. Ein Ausweg stellt ein fehlersicherer Systemkern (Hardcore) dar.
Ein Modell, in dem die Einheiten nicht logisch unterschieden sind, heißt dann nicht hierarchisch. Ein 2-Phasenmodell, das Merkmal a) erfüllt, wird in [60] vorgestellt, eine abstrakte Definition eines hierarchischen Modells in [42] gegeben. Ein Beispiel für hierarchische Diagnose mit Merkmal b) wäre die traditionelle Diagnose eines Mikrocomputers. Als Einheiten werden der Systemkern, die Register, die ROM- und RAM-Speicher und die Input/Output-Interfaces in dieser Reihenfolge jeweils unter Benutzung der Diagnose der voranstehenden Einheiten diagnostiziert.

Die Verwendung einer zentralen Diagnose mit globaler Sicht verlangt nach Ausfallsicherheit der einen Diagnoseinstanz. Dezentrale Diagnose verteilt das Risiko von defekten Diagnoseinstanzen auf mehrere Einheiten. Das Modell einer dezentralen, nicht hierarchischen Diagnose mit globaler Sicht wird in [44] vorgestellt. Diagnose mit lokaler Sicht verwendet das Tübinger System ATTEMPTO (vergleiche Kapitel 6).

Anmerkungen:

1) Nicht alle Parameter in 2.2.1. - 2.2.3. sind unabhängig voneinander. So stellt zum Beispiel eine hierarchische Diagnose notwendigerweise eine Mehrschrittdiagnose dar oder eine zentrale Diagnose bedarf einer globalen Sicht.

2) Der Sprachgebrauch der Parameter in 2.2.3. ist in der Literatur nicht einheitlich. Eine dezentrale Diagnose mit globaler Sicht wird oft auch als verteilte Diagnose bezeichnet (etwa in [44]).

3) Bei der Festlegung der Testmenge in 2.2.2. muß beachtet werden, daß Nachrichtenaustausch in einem System nur über die vorhandenen (physikalischen) Verbindungswege möglich ist. Ein

Test für eine Menge von Einheiten darf also nur vorgesehen werden, wenn die benötigten Kommunikationswege im System vorhanden sind. Liegt ein Netzwerk von Einheiten vor, ist ein Kommunikation auch zwischen Einheiten, die physikalisch nicht miteinander verbunden sind, möglich, wenn weitere Einheiten Vermittlungsfunktionen übernehmen. Probleme machen dann defekte Einheiten, die die zu vermittelnden Nachrichten verfälschen können ([65,46,47,28]). Einen Überblick über bekannte Netzstrukturen von Einheiten gibt [30], neuere Arbeiten betrachten die auf den Moore-Graphen aufgebauten regelmäßigen Netze [7], binäre Bäume oder Variationen dieser [39] sowie die für Fehlerdiagnose besonders geeigneten Boolschen n-Kuben [8] und die fehlertoleranten Vernetzungen in [68].

## 2.3. DIE GÜTE VON DIAGNOSESTRUKTUREN

Wird eine Diagnosestruktur dynamisch aufgebaut, hängt sie vom konkret vorliegenden Fehlerzustand des System ab und besitzt kein allgemeines Fehlerdiagnosevermögen. Die Güte von Diagnosestrukturen bezüglich Fehlerdiagnose läßt sich nur bei statischer Diagnose bewerten. Im folgenden sollen zunächst bekannte Maße für diese Bewertung vorgestellt werden, danach mögliche neue Maße vorgeschlagen.

a) bekannte Maße:

Eine Diagnosestruktur heißt t-diagnostizierbar, falls aus den Ergebnissen der in ihr vorgesehenen Tests stets eine korrekte und vollständige Diagnose gelingt, wenn nicht mehr als t Einheiten defekt sind [69]. Treten mehr als t defekte Einheiten auf, kann eine Diagnose unmöglich werden oder auch inkorrekt.
Eine Struktur heißt t-detektierbar, falls bei höchstens t defekten Einheiten mit Hilfe der Testergebnisse stets entschieden werden kann, ob defekte Einheiten vorliegen oder nicht [74].
Da Fehlerdetektierung im allgemeinen einfacher ist als Fehlerdiagnose, erscheint die folgende Art einer zweigeteilten Diagnose sehr sinnvoll: Im ersten Diagnoseteil wird durch Fehlerdetektierung festgestellt, ob überhaupt defekte Einheiten vorliegen. Nur

wenn dies der Fall ist, wird im zweiten Teil die kompliziertere Fehlerdiagnose, also die Lokalisierung der Fehler, durchgeführt. Genaue graphentheoretische Charakterisierungen der t-Diagnostizierbarkeit und der t-Detektierbarkeit von Diagnosestrukturen werden für einige Modelle in den Kapiteln 3 und 4 gegeben.
Falls auch intakte Einheiten als defekt diagnostiziert werden dürfen, bietet sich das von Friedman [31] angegebene Maß der s-t-Diagnostizierbarkeit an: Eine Diagnosestruktur heißt dabei s-t-diagnostizierbar ($s \geq t$), falls bei Auftreten von höchstens t Fehlern eine Menge von höchstens s Einheiten angegeben werden kann, die alle defekten Einheiten enthält. Alle diese s Einheiten werden als defekt angesehen. Graphentheoretische Charakterisierungen s-t-diagnostizierbarer Diagnosestrukturen existieren (noch) nicht, nur Aussagen über spezielle Strukturen wie Diagnoseringe [31].
Ein geeignetes Maß bei Mehrschrittdiagnose stellt die sogenannte sequentielle t-Diagnostizierbarkeit [69] dar. Ein sequentiell t-diagnostizierbarer Diagnosegraph muß unter der Voraussetzung, daß zwischen einer und t Einheiten defekt sind, stets die Erkennung wenigstens einer defekten Einheit garantieren. Besteht ein System aus n Einheiten, so gelingt also die vollständige Diagnose in höchstens n Diagnoseschritten, falls als defekt erkannte Einheiten repariert werden und während der Diagnose nicht mehr ausfallen. Ansätze zur Charakterisierung sequentiell t-diagnostizierbarer Diagnosestrukturen wurden in [19,54,59] gemacht.
Während alle bisher angeführten Maße auf der t-Fehlerannahme aufbauen, beruht das folgende auf probabilistischer Diagnose:
Eine Diagnosestruktur heißt probabilistisch r-diagnostizierbar ($0 \leq r < 1$, $r \in \mathbb{R}$) oder kurz p-r-diagnostizierbar [55], falls zu jeder Familie von Testergebnissen stets höchstens eine mit den Testergebnissen verträgliche Fehlermenge mit Wahrscheinlichkeit $> r$ existiert ( Die Wahrscheinlichkeit einer Fehlermenge wird dabei aus den Einzelausfallswahrscheinlichkeiten ihrer Elemente berechnet). In [55] wird zu dem dort vorgestellten probabilistischen Modell auch ein Charakterisierungssatz für p-t-diagnostizierbare Diagnosegraphen angegeben.
Das letzte angegebene Maß beschreibt den Zeitbedarf, der für den Aufbau einer Diagnosestruktur, d.h. für die Ausführung aller vorgesehenen Tests, notwendig ist. Zu einem Zeitpunkt durchge-

führte Tests bilden dabei eine Testrunde. Man beachte, daß eine Einheit zu einem Zeitpunkt nur an einem Test beteiligt sein kann. Eine Diagnosestruktur heißt k-konstruierbar, falls alle in ihr vorgesehenen Tests in k Testrunden durchgeführt werden können. Ist die Diagnosestruktur sogar ein Diagnosegraph, so stimmen die Begriffe Konstruierbarkeit und Kantenfärbbarkeit [64] überein. Die Konstruierbarkeit einer Diagnosestruktur ist also ein Maß für die Parallelisierbarkeit des Testeinsatzes. Berechnungen der Zahl von Testrunden werden in [53]für das dort verwendete Modell angestellt; in Kapitel 4 wird die Frage der Konstruierbarkeit für die Vergleichstestmodelle angesprochen.
Man beachte, daß der Begriff der k-Konstruierbarkeit auch für dynamisch aufgebaute Diagnosestrukturen sinnvoll sein kann: Eine dynamische Diagnosestrategie ist k-konstruierbar, falls sie im ungünstigsten Fall (worst case) mit k Testrunden auskommt.

b) mögliche neue Maße:

Die bisher aufgeführten Maße sprechen nur von Einheiten des Systems und von Fehlern in Einheiten. Diese Angaben lassen sich verfeinern.

i) Unterscheidung der Einheiten in aktive Einheiten (z.B. Prozessoren), passive Einheiten (z.B. Speicher) und Kommunikationseinheiten (Links,Busse). Statt der pauschalen t-Fehlerannahme kann dann eine $(t_1,t_2,t_3)$-Fehlerannahme gemacht werden. Höchstens $t_1$ aktive, $t_2$ passive und $t_3$ Kommunikationseinheiten dürfen ausfallen. Dies motiviert eine $(t_1,t_2,t_3)$-Diagnostizierbarkeit als Maß einer Diagnosestruktur.

ii) Unterscheidung der zugelassenen Fehler.

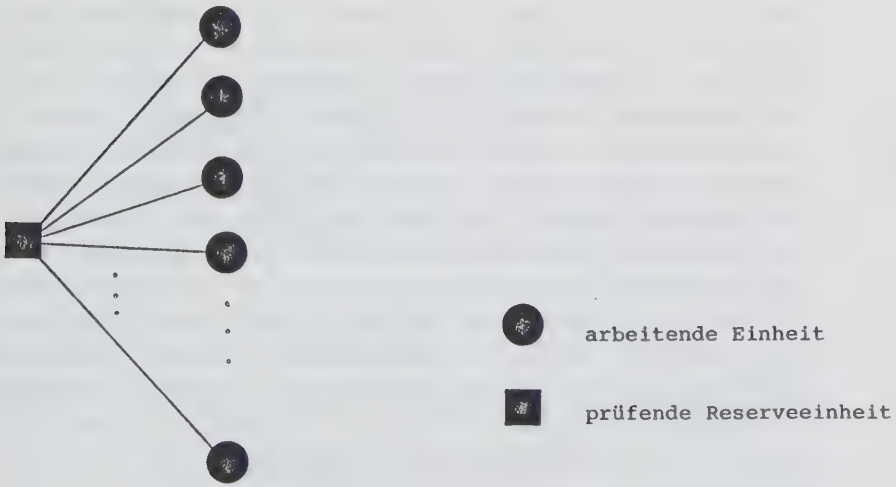
Eine Diagnosestruktur heißt $(s_1,s_2)$-diagnostizierbar, falls bei Auftreten von höchstens $s_1$ permanenten und höchstens $s_2$ transienten Fehlern die betreffenden Einheiten und die in ihnen aufgetretene Fehlerart erkannt werden kann.

## 2.4. BEISPIELE EXISTIERENDER SYSTEME UND BEKANNTER MODELLE

In diesem Abschnitt sollen zunächst die beiden Systeme MICRONET und SIFT in das durch die Abschnitte 2.1. und 2.2. gegebene Parametersystem eingefügt werden und einige weitere Systeme kurz beschrieben. Anschließend werden die wichtigsten theoretischen Modelle der Selbstdiagnose, die in der Literatur auftauchen, beschrieben.

a) Systeme:

MICRONET (nach [72]): Die Struktur und Arbeitsweise von MICRONET wurde bereits erklärt. Im Bereich eines Strings von Computern ergibt sich als Diagnosestruktur die 1-diagnostizierbare Sternstruktur (Figur 2.2.).




Figur 2.2. Die Diagnosestruktur eines Strings in MICRONET

Hier nun die Einordnung von MICRONET in das Selbstdiagnoseparametersystem:

2.1.1. - Computer
- stochastisch unabhängig
- gleichwahrscheinlich
2.1.2. - alle Fehler
2.1.3. - disjunkte 1-1-Tests (d.h. s=r=1)
- m=2
- ungerichtet
- vollständig
2.1.4. - funktionsbegleitend

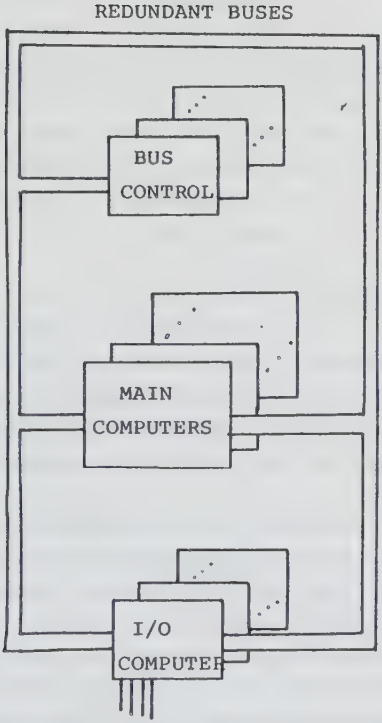
2.2.1. - 1-Fehlerannahme
- keine Ersetzung intakter Einheiten (Einheiten, in denen transiente Fehler entdeckt wurden, werden allerdings als defekt angesehen)
- 1-Schrittdiagnose
- (keine weiteren Testannahmen)
2.2.2. - statisch (Sternstruktur)
- keine Mehrfachanwendung von Tests
2.2.3. - zentral
- global
- nicht hierarchisch.

Man beachte dabei, daß der Bereich der Diagnose jeweils nur ein String von Computern und nicht das gesamte System ist.

SIFT (nach [78]): SIFT ( = in Software implementierte Fehlertoleranz) ist ein Computersystem zur aktiven Steuerung von Flugzeugen (entwickelt am SRI). Die hohe Anforderung einer Ausfallswahrscheinlichkeit von höchstens $10^{-9}$/Stunde in einem 10-Stundenflug soll SIFT für Fluganwendungen geeignet machen, in denen die Flugsicherheit entscheidend vom ständigen Funktionieren des Computers abhängt ( in heute gebräuchlichen Flugzeugen können die Funktionen des Bordcomputers etwa von der Besatzung übernommen werden). Fehlertoleranz wird in SIFT durch eine Vervielfachung von Systemaufgaben auf mehrere Prozessoren und durch anschließendes (software-mäßiges) Voten erreicht.Die Prozessoren arbeiten asynchron und die Kommunikationsbusse sind redundant ausgelegt. Die Struktur von SIFT zeigt Figur 2.3. (aus [72]).

Hier die Selbstdiagnose-Parameter von SIFT:

2.1.1. - Computer und Busse
- stochastisch unabhängig
- nicht gleichwahrscheinlich (verschiedene Arten von Einheiten)
2.1.2. - alle Fehler
2.1.3. - disjunkte 1-1-Tests
- m=2
- ungerichtet
- vollständig
2.1.4. - funktionsbegleitend




Figur 2.3. Die SIFT-Struktur

2.2.1. - t-Fehlerannahme ( t= ⌊(n-1)/2⌋ bei Ver-n-fachung des Programms)
- keine Ersetzung intakter Einheiten
- 1-Schrittdiagnose

2.2.2. - statisch (Majoritätsbilding)
- keine Mehrfachanwendung von Tests

2.2.3. - zentral
- global
- nicht hierarchisch

Einige Beispiele weiterer fehlertoleranter Systeme sind das MuTEAM-System [20], das DIRMU-Baukastensystem [52]und das RDC-System [76]. MuTEAM ist ein Prototyp-System für die

Entwicklung von integrierten Designmethoden für verteilte Multi-Mikroprozessor-Systeme mit den Hauptzielen Modularität, Erweiterbarkeit, Fehlerbegrenzung und Robustheit. Ein Baukastensystem, das je nach Anwendung geeignet konfiguriert werden kann, wird durch DIRMU ( = Distributed Reconfigurable Multiprocessor Kit) angeboten. Die Selbstdiagnosestrategie ist dabei unabhängig von der konkreten Konfiguration. RDC ist ein fehlertolerantes, verteiltes Mikrorechnersystem mit Lichtleiterkopplung, das zur Steuerung von Tieföfen konstruiert wurde. Als Diagnosestruktur benutzt RDC einen (bidirektionalen) Ring.

b) einige bekannte Modelle:

Das Modell von Preparata, Metze und Chien (PMC-Modell) [69]:
Dies ist das klassische Modell der (Selbst-) Diagnose und wird in Kapitel 3 ausführlich beschrieben. Als erstes bekanntes theoretisches Modell hat es seit seiner Publizierung 1967 eine Fülle von Untersuchungen auf dem Gebiet der Selbstdiagnose und Fehlertoleranz ausgelöst. Wichtigste Merkmale dieses Modells sind Tests, von denen genau eine testende und eine getestete Einheit betroffen sind, und die Annahme, daß fehlerhafte testende Einheiten in ihrem Verhalten nicht beschränkt werden.

Das Modell von Friedman [31]:
Wichtigstes Kennzeichen dieses Modells ist es, daß auch eine gewisse Anzahl fehlerfreier Einheiten als defekt diagnostiziert werden darf. Diese Annahme erscheint insofern gerechtfertigt, als daß in der Praxis oft nicht genau die Menge der fehlerhaften Einheiten bestimmt werden kann, sondern nur eine (meist größere) Menge von Einheiten, unter deren Elementen alle fehlerhaften Einheiten vermutet werden.

Das Modell von Maheshwari und Hakimi [55]:
Abgesehen von der Annahme der probabilistischen Fehlerdiagnose macht dieses Modell dieselben Annahmen wie das PMC-Modell. Zu beachten ist auch, daß bei der Auswertung der Testergebnisse die Ausfallswahrscheinlichkeiten der einzelnen Einheiten eine Rolle spielen.

In allen diesen drei Modellen treten (gewöhnliche) Graphen als Diagnosestrukturen auf. Das folgende Modell benötigt (gerichtete)

Hypergraphen.

Das Modell von Russell und Kime [74,75]:
Dies stellt eine Verallgemeinerung des PMC-Modells dar. Die Unterschiede liegen dabei in der Verwendung von disjunkten s-1-Tests, d.h. s Einheiten testen gemeinsam eine weitere Einheit, und in den Annahmen über Testergebnisse.
Morphische Version: Ein Test wird invalidiert, d.h. das Ergebnis ist nicht vertrauenswürdig, falls auch nur eine der Tester-Einheiten defekt ist.
Semimorphische Version: Durch einen Boolschen Ausdruck wird angegeben, bei welcher Konstellation in der Testermenge der Test invalidiert wird.
Die Motivation für Werte s>1 kommt daher, daß zur Durchführung eines Tests verschiedene Betriebsmittel eines Systems ( etwa ein Prozessor, ein Speicher und eine Kommunikationsverbindung) benötigt werden.

Die Modelle von Mallela und Masson [57,58]:
Aufbauend auf den Grundannahmen des PMC-Modells werden hier Aussagen über die Eigenschaften von Diagnosestrukturen gemacht, Mengen von intermittierenden [57] bzw. gemischte Mengen von intermittierenden und permanenten Fehlern [58] zu diagnostizieren.

# 3. DIE KLASSISCHEN MODELLE DER SELBSTDIAGNOSE

In diesem Kapitel soll eine Klassifizierung und Untersuchung der sogenannten klassischen Modelle der Selbstdiagnose vorgenommen werden. Klassisch werden dabei Modelle genannt, für die die folgenden Parameter bereits festgelegt sind.

Tests:
- 2.1.1. aktive Einheiten (z.B. Prozessoren), Ausfälle sind stochastisch unabhängig
- 2.1.2. nur permanente Fehler
- 2.1.3. disjunkte 1-1-Tests, m=2, gerichtete Tests

Diagnose:
- 2.2.1. t-Fehlerannahme, intakte Einheiten dürfen nicht ersetzt werden, 1-Schrittdiagnose
- 2.2.2. statische Diagnose (Verwendung von Diagnosegraphen, keine Mehrfachanwendung von Tests

Maße für die Diagnosestrukturen: t-Diagnostizierbarkeit und t-Detektierbarkeit.

Die nicht aufgeführten Parameter Ausfallswahrscheinlichkeit der Einheiten und Organisation des Testeinsatzes und der Auswertung der Testergebnisse werden nicht festgelegt, da sie als implementierungsabhängig angesehen werden und erst bei einer Realisierung eines Modells in einem Computersystem festgelegt werden. Die verschiedenen klassischen Modelle unterscheiden sich also nur in der Wahl von vollständigen oder unvollständigen Tests und in den Annahmen über die Testergebnisse. Man beachte, daß die Beschränkung auf disjunkte 1-1-Tests (d.h. $|A| = |B| = 1$, A und B disjunkt) und das Ausschließen von Mehrfachanwendungen von Tests dazu führt, daß die jeweils vorliegende Diagnosestruktur ein gerichteter (Diagnose-) Graph ohne Schlingen (d.h. Kanten mit identischem Anfangs- und Endknoten) und ohne Mehrfachkanten (d.h. Kanten mit identischem Paar von inzidenten Knoten) wird, also ein "Digraph" im Sinne von [36].

Der folgende Abschnitt 3.1. liefert die nötigen Bezeichnungen und Definitionen für das weitere Vorgehen, außerdem wird die Menge möglicher klassischer Modelle auf im wesentlichen 10 verschiedene Typen reduziert. Bezüglich der benutzten Maße t-Diagnostizierbarkeit und t-Detektierbarkeit erweist sich dabei einer dieser Typen als trivial und uninteressant, da er niemals auch nur geringe Fehlererkennungsmöglichkeiten garantiert. Die anderen 9 Modelle

(Vertreter der 9 anderen Typen) sind Untersuchungsgegenstand von Abschnitt 3.2. (bezüglich t-Diagnostizierbarkeit) und Abschnitt 3.3. (bezüglich t-Detektierbarkeit). Abschnitt 3.4. schließlich fragt nach logischen Beziehungen zwischen den Modellen. Dies führt zu Verbandsstrukturen auf der Menge der klassischen Modelle der Selbstdiagnose.

## 3.1. BEZEICHNUNGEN UND REDUKTIONEN

Ein System S bestehe aus einer Menge $N=\{u_1,\ldots,u_n\}$ ($n\geq 3$) von n Einheiten; die Menge der durchzuführenden Tests sei $E = \{(u_i,u_j) \mid u_i,u_j\in N,\ u_i \text{ testet } u_j\}$. Diese Situation läßt sich durch einen endlichen, gerichteten Graphen $G=(N,E)$ darstellen, in dem N die Knotenmenge und E die Kantenmenge bildet. G heißt der Diagnosegraph des Systems S. Nach den Parameterannahmen für klassische Modelle ist E eine Menge (kein Mehrfachauftreten einer Kante $(u_i,u_j)$) und es gilt $E\cap\{(u_i,u_i)\mid i=1,2,..,n\}=\emptyset$. Dagegen ist die Situation $\{(u_i,u_j),(u_j,u_i)\}\subseteq E$ für $i\neq j$ zugelassen, es liegt dann ein Doppeltest zwischen $u_i$ und $u_j$ vor. Im Unterschied dazu heißt ein Test $(u_i,u_j)\in E$ mit $(u_j,u_i)\notin E$ ein Einzeltest.
Die beiden möglichen Zustände werden mit $1\leftrightarrow$ intakt (fehlerfrei) und $0\leftrightarrow$ defekt (fehlerhaft) bezeichnet, die beiden möglichen Ausgänge eines Tests mit $1\leftrightarrow$"kein Fehler gefunden" und $0\leftrightarrow$"Fehler gefunden". Unter einem Syndrom wird eine vollständige Familie von Testergebnissen verstanden, was einer vollständigen Kantenbewertung des Diagnosegraphen mit Werten aus {0,1} entspricht. Ein Fehlermuster ist eine Menge von defekten Einheiten; es ist konsistent mit einem Syndrom, falls es dieses bewirken kann.

Die Testannahmen eines klassischen Modells werden durch Angabe eines Quadrupels $(t_{11},t_{10},t_{01},t_{00})$ festgelegt, wobei $t_{ij}$ $(i,j\in\{0,1\})$ den Testausgang eines Tests angibt, bei dem sich die testende Einheit im Zustand i und die getestete Einheit im Zustand j befindet. Man beachte, daß Vollständigkeit eines Tests äquivalent ist mit $t_{11}=1$ und $t_{10}=0$. Im folgenden werden wir stets annehmen, daß ein Test einer intakten Einheit durch eine andere intakte Einheit den Testausgang $t_{11}=1$ ergibt. Dagegen können für $t_{10}$, $t_{01}$

und $t_{00}$ jeweils die Ausgänge 1, 0 oder X vorgeschrieben werden (X bedeutet, daß beide Ausgänge 1 oder 0 zugelassen werden).

Definition (Spiegelung von klassischen Modellen):
Seien A und B klassische Modelle mit den zugehörigen Quadrupeln $(t^Y_{11}, t^Y_{10}, t^Y_{01}, t^Y_{00})$ (Y=A,B). Dann heiße A die Spiegelung von B (in Zeichen $A=B^{sp}$), falls $t^A_{10}=t^B_{01}$, $t^A_{01}=t^B_{10}$ und $t^A_{00}=t^B_{00}$ gilt.

Diese Definition ist symmetrisch, d.h. aus $A=B^{sp}$ folgt sofort $B=A^{sp}$.

Definition (Inversion von gerichteten Graphen):
Sei G=(N,E) ein gerichteter Graph. Ein gerichteter Graph $G'=(N',E')$ heiße die Inversion von G (in Zeichen $G'=G^{inv}$), falls N'=N und $E'=\{(u_i,u_j) \mid u_i,u_j \in N' \text{ und } (u_j,u_i)\in E\}$ gilt.
Die Inversion eines Graphen entsteht also durch Umkehr der Richtungen aller Kanten des Graphen. Auch diese Definition ist symmetrisch, ein zweimalige Anwendung der Operation $^{inv}$ führt zum Ausgangsgraphen zurück.

Definition (Vereinfachung von gerichteten Graphen):
Ein Graph G' heißt eine Vereinfachung eines Graphen G, falls G' aus G durch Ersetzen aller Doppeltests durch jeweils einen der möglichen Einzeltests entsteht.
In G' gilt also $\left[(u_i,u_j)\in E' \Longrightarrow (u_j,u_i)\notin E'\right]$ .

Definition (Doppeltestkomponente eines gerichteten Graphen):
Sei G=(N,E) ein gerichteter Graph ohne Schlingen und Mehrfachkanten. Der Graph G' entstehe aus G, in dem aus E alle Kanten $(u_i,u_j)$ entfernt werden, für die $(u_j,u_i)\notin E$ gilt. Dann heiße eine Teilmenge $X\subseteq N$, die die Knotenmenge einer starken Zusammenhangskomponente von G' [36] ist, eine Doppeltestkomponente von G. Die Menge aller Doppeltestkomponenten von G wird mit DTK(G) bezeichnet.

Hier noch einige graphentheoretische Definitionen und Bezeichnungen. Dabei sei stets G=(N,E) ein gerichteter Graph ohne Schlingen und Mehrfachkanten.
$N(u):=\{u_i \mid u_i\in N,\ (u,u_i)\in E\}$ Nachfolgermenge von $u\in N$.
$V(u):=\{u_i \mid u_i\in N,\ (u_i,u)\in E\}$ Vorgängermenge von $u\in N$.

$N(U) := \bigcup_{u \in U} N(u) \setminus U$ (echte) Nachfolgermenge von $U \subseteq N$.

$V(U) := \bigcup_{u \in U} V(u) \setminus U$ (echte) Vorgängermenge von $U \subseteq N$.

$d_{in}(u) := |V(u)|$ Eingangsgrad von $u \in N$ .
$d_{out}(u) := |N(u)|$ Ausgangsgrad von $u \in N$ .
$d_{in,min} := \min\{d_{in}(u) \mid u \in N\}$ minimaler Eingangsgrad von G.

Im folgenden seinen alle betrachteten Diagnosegraphen (schwach) zusammenhängend [36]. Dies ist eine sinnvolle Einschränkung, da im anderen Falle jede einzelne Zusammenhangskomponente als selbständiger Diagnosegraph behandelt werden könnte. Zu den festen Annahmen der klassischen Modelle gehört die t-Fehlerannahme, stets sei $t \geq 1$. Die im folgenden benutzten Maße t-Diagnostizierbarkeit und t-Detektierbarkeit sind wie folgt definiert.

Definition:
Sei S ein System mit Diagnosegraph G, sei A das benutzte Modell für die Selbstdiagnose und sei $t \geq 1$.
Dann heiße S bzw. G t-diagnostizierbar in Modell A, falls unter der Voraussetzung, daß die Annahmen des Modells A gelten und daß nicht mehr als t Einheiten defekt sind, aus der Kenntnis des Syndroms heraus die Menge aller defekten Einheiten erschlossen werden kann.
S bzw. G heiße t-detektierbar in Modell A, falls unter der Voraussetzung der Annahmen des Modells A und von nicht mehr als t defekten Einheiten aus der Kenntnis des Syndroms heraus erschlossen werden kann, ob defekte Einheiten vorliegen.
Schließlich heiße S bzw. G t-optimal in Modell A, falls S bzw. G t-diagnostizierbar in Modell A ist und unter allen t-diagnostizierbaren Systemen bzw, Graphen in Modell A mit gleicher Knotenzahl eine minimale Kantenzahl besitzt.

Nach der Festlegung von $t_{11}=1$ gibt es 9 verschiedene klassische Modelle, die vollständige Tests (d.h. $t_{10}=0$) vorsehen. In diesem Fall sind nämlich die beiden Quadrupeleinträge $t_{01}$ und $t_{00}$ frei aus {0,1,X} zu wählen. Im folgenden wird nun gezeigt, daß bis auf uninteressante Modelle, die keinerlei Fehlererkennungsmöglichkeit bieten, bis auf geeignete Uminterpretierung diese 9 Modelle bereits alle interessanten Möglichkeiten ausschöpfen.

Lemma 3.1.:
Sei A ein klassisches Modell für die Selbstdiagnose.
Gilt $\{t^A_{10}, t^A_{01}\} \subseteq \{1,X\}$, so ist in Modell A niemals t-Diagnostizierbarkeit oder t-Detektierbarkeit für ein $t \geq 1$ möglich.

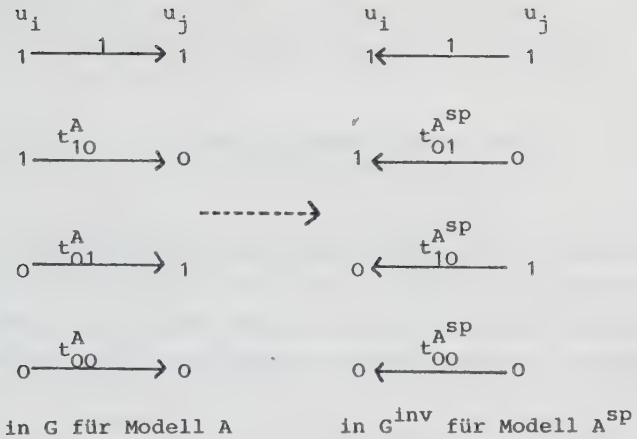
Beweis: Bei Auftreten genau einer defekten Einheit ist immer das 1-Syndrom, d.h. das Syndrom mit ausschließlich Testergebnissen 1, möglich. Dieser Zustand kann dann mit dem völlig intakten System verwechselt werden. q.e.d.

Interessante Modelle müssen also stets die Bedingung $\left[t_{10}=0 \vee t_{01}=0\right]$ erfüllen. 9 dieser Modelle, die $t_{10}=0$ verlangen, wurden schon erwähnt; die restlichen 6 Modelle, für die $t_{10} \in \{1,X\}$ und $t_{01}=0$ verlangt wird, werden im folgenden Lemma als nur unwesentlich verschiedene Varianten dieser Modelle erkannt.

Lemma 3.2.:
Sei A ein klassisches Modell und G ein Diagnosegraph.
Dann ist G genau dann t-diagnostizierbar in Modell A, wenn $G^{inv}$ t-diagnostizierbar in Modell $A^{sp}$ ist. Die entsprechende Aussage gilt für die t-Detektierbarkeit.

Beweis: Da die beiden Operationen Spiegelung von Modellen und Inversion von Graphen symmetrisch sind, genügt es, jeweils eine Implikationsrichtung der beiden Aussagen zu beweisen.
t-Diagnostizierbarkeit: Die folgende Aufstellung zeigt, daß ein Fehlermuster stets sowohl in G für Modell A als auch in $G^{inv}$ für Modell $A^{sp}$ dasselbe Syndrom hervorrufen kann (wenn die Reihenfolge der Tests bei der Inversion beibehalten wird) (Figur 3.1.).
Man beachte $t^A_{10}=t^B_{01}$, $t^A_{01}=t^B_{10}$ und $t^A_{00}=t^B_{00}$ (wobei $B=A^{sp}$).
Ist also G nicht t-diagnostizierbar in Modell A und bewirken die verschiedenen Fehlermuster $F_1$ und $F_2$, die beide höchstens t Elemente enthalten, dasselbe Syndrom in G und Modell A, so sind $F_1$ und $F_2$ dazu auch in $G^{inv}$ und Modell $A^{sp}$ in der Lage. Also ist auch $G^{inv}$ in Modell $A^{sp}$ nicht t-diagnostizierbar.

t-Detektierbarkeit: Ist G nicht t-detektierbar in Modell A, so existiert ein Fehlermuster F mit höchstens t Elementen, das das 1-Syndrom hervorrufen kann. Nach obigen Überlegungen ist dann F auch in $G^{inv}$ und Modell $A^{sp}$ dazu in der Lage. q.e.d.




Figur 3.1. Übergang von (G,A) zu $(G^{inv},A^{sp})$.

Lemma 3.2. zeigt, daß die noch fraglichen Modelle A mit $t^A_{10}\in\{1,X\}$ und $t^A_{01}=0$ im wesentlichen nur Uminterpretationen ihrer Spiegelungen $A^{sp}$ sind.

In die nähere Untersuchung der Abschnitte 3.2. und 3.3. können wir deshalb mit den 9 Modellen mit $t_{10}=0$ gehen, ohne deswegen interessante Modelle auszuklammern.
Hier eine Auflistung und Bezeichnung der 9 Modelle (Figur 3.2.).

| | Modell | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | PMC | BGM | A1 | S1 | S2 | S3 | A2 | A3 | A4 |
| $t_{11}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $t_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_{01}$ | X | X | X | 0 | 0 | 0 | 1 | 1 | 1 |
| $t_{00}$ | X | 0 | 1 | X | 0 | 1 | X | 0 | 1 |

Figur 3.2. Die 9 interessanten klassischen Modelle

Beim ersten aufgeführten Modell steht PMC für die Urheber des Modells [69], ebenso wie BGM im zweiten Modell [11]. Die übrigen 7 Modelle tauchen in der Literatur noch nicht auf.

## 3.2. UNTERSUCHUNG DER KLASSISCHEN MODELLE BEZÜGLICH T-DIAGNOSTIZIERBARKEIT

Unter einer Untersuchung eines Modells wird ein Eingehen auf folgende Punkte verstanden:

- Angabe eines Struktursatzes, d.h. einer graphentheoretischen Charakterisierung der t-Diagnostizierbarkeit im jeweiligen Modell.
- Angabe eines Strukturalgorithmus, d.h. eines Algorithmus, der den höchsten möglichen Grad t der Diagnostizierbarkeit eines vorgegebenen Diagnosegraphen bestimmt.
- Präsentation von t-optimalen Diagnosegraphen.
- Angabe eines Diagnosealgorithmus, d.h. eines Algorithmus, der für einen gegebenen Diagnosegraphen aus einem vorliegenden Syndrom das aufgetretene Fehlermuster bestimmt. Bei Verwendung eines t-diagnostizierbaren Diagnosegraphen und mit der Annahme, daß höchstens t Fehler aufgetreten sind, bedeutet dies also, das eindeutige, zum Syndrom konsistente Fehlermuster mit höchstens t Elementen zu finden.
- Fragen der Realisierbarkeit des Modells in Computersystemen (Validierung).

i) Das PMC-Modell:

Dieses erste theoretische Modell der Diagnose in Computersystemen wurde 1967 von Preparata, Metze und Chien vorgestellt [69]. Von allen 9 klassischen Modellen macht es die schwächsten Modellannahmen, da die beiden Quadrupeleinträge $t_{01}$ und $t_{00}$ jeweils beide Testausgänge zulassen.

SATZ 3.1. [35]
Sei G=(N,E) ein Diagnosegraph mit $|N|=n$.
Dann ist G t-diagnostizierbar im PMC-Modell genau dann, wenn gilt 1) $n \geq 2t+1$

2) $d_{in,min} \geq t$

3) $\forall 0 \leq p < t \; \forall U \subseteq N \left[ |U| = n-2t+p \Rightarrow |N(U)| > p \right]$ .

Eine Charakterisierung der t-Diagnostizierbarkeit, die auf einer 3-Partition der Einheitenmenge beruht, wird in [3] angegeben. Ein effektiver Strukturalgorithmus für das PMC-Modell, der also in polynomialer Zeit den höchsten möglichen Grad t der Diagnostizierbarkeit liefert, ist nicht bekannt. Eine grobe untere Schranke für den Wert t gibt

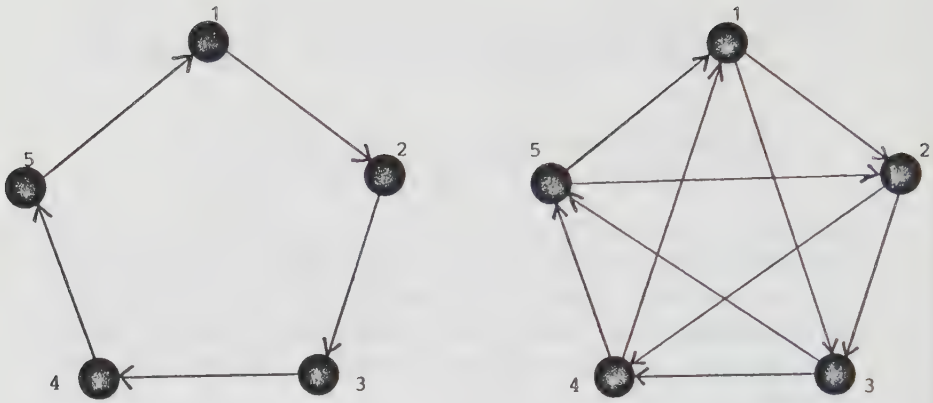
Lemma 3.3.:
Sei G=(N,E) ein Diagnosegraph und sei $t := \lfloor d_{in,min}/2 \rfloor \geq 1$.
Dann ist G t-diagnostizierbar im PMC-Modell.

Beweis: Eine eindeutige Diagnose des Zustands jeder Einheit $u \in N$ gelingt in der folgenden Weise. Sicher ist $|V(u)| \geq d_{in,min} \geq 2t$. Ist u intakt, so ergeben mindestens die t Tests $(u_i,u) \in E$ mit $u_i \in V(u)$ und $u_i$ intakt das Testergebnis 1. Ist dagegen u defekt, so können allenfalls die höchstens t-1 defekten Elemente aus V(u) das Testergebnis 1 über u bewirken. q.e.d.

Die Bestimmung des minimalen Eingangsgrades eines Graphen kommt mit $O(n^2)$ bzw. $O(|E|)$ Schritten aus, je nach Darstellung des Graphen durch eine Adjazenzmatrix oder eine Adjazenzliste.

In [69] werden für alle nach Satz 3.1. möglichen Parameter n und t die Graphen $D_{1t}$ mit n Knoten als t-optimale Diagnosegraphen erkannt. Dabei werden die Einheiten von 1 bis n durchnumeriert und die i-te Einheit testet die t Einheiten mit den Nummern i+1,i+2,...,i+t (modulo n) ($1 \leq i \leq n$). Beispiele solcher Graphen zeigt Figur 3.3..

Große Aufmerksamkeit wurde der Auffindung eines Diagnosealgorithmus für das PMC-Modell geschenkt ([22,40,44,50,51,62,63]).
Dabei benutzt der Algorithmus in [40]das "Backtracking"-Verfahren,




Figur 3.3. Beispiele von $D_{1t}$-Graphen: $D_{11}$ und $D_{12}$ mit jeweils 5 Knoten

d.h. Einheiten werden zunächst vorläufig diagnostiziert und diese Entscheidung später eventuell wieder revidiert. Allerdings ist dieser Algorithmus nicht ganz korrekt (Gegenbeispiel $D_{12}$ mit 5 Knoten). Corluhan und Hakimi [22] geben einen Algorithmus an, der unter der Voraussetzung, daß der Diagnosegraph der folgenden Hypothese genügt, korrekt arbeitet.

Hypothese von Corluhan/Hakimi:

Für einen t-diagnostizierbaren Graphen G=(N,E) und bei Auftreten von höchstens t Fehlern gibt es ein u∈N, so daß gilt

$$\left[\,|\{v\in N \mid u \text{ intakt} \Rightarrow v \text{ defekt}\}| > t \;\vee\; |\{v\in N \mid u \text{ defekt} \Rightarrow v \text{ defekt}\}| > t\right].$$

Im ersten Fall wird dann u als defekt, im zweiten als intakt diagnostiziert.

In [50] wird allerdings ein Beispielgraph gegeben, der dieser Hypothese nicht genügt, so daß dieser Diagnosealgorithmus nur für eine Teilklasse aller Diagnosegraphen anwendbar ist.

Ebenfalls auf eine Teilklasse von Diagnosegraphen, nämlich die oben definierten $D_{1t}$-Graphen, beschränkt sich der in [62] beschriebene Diagnosealgorithmus, der in $O(n^2)$ Schritten auskommt. Erweitert wurde dieses Resultat auf eine größere Klasse von Graphen in [63]. In diese Klasse fallen alle t-diagnostizierbaren Graphen mit $t \leq 25$. Kuhl und Reddy beschreiben in [44] einen

verteilten Diagnosealgorithmus für das PMC-Modell. Jede Einheit entwickelt dabei eine globale Diagnosesicht über das System. Eine Entscheidung darüber, welche Einheiten aber die korrekte Systemsicht entwickelt haben, ist hier nicht möglich und auch nicht vorgesehen.
Als bisher einziger zentral organisierter Diagnosealgorithmus, der in polynomial vielen Schritten auskommt, ist der Algorithmus von Madden [50,51] bekannt. Unter Verwendung des bekannten Warshall-Algorithmus [84] gelingt die Diagnose in $O(n^3)$ Schritten bei allerdings sehr hohem Speicherplatzbedarf.

Eine Realisierung des PMC-Modells, wie auch der anderen 8 klassischen Modelle, hängt entscheidend von der Implementierung vollständiger (oder wenigstens nahezu vollständiger) Tests ab.

ii) Das BGM-Modell:
Das BGM-Modell macht im Unterschied zum PMC-Modell die Annahme, daß beim Test einer defekten Einheit sich stets das Testergebnis O ergibt, daß also stets ein Fehler gefunden wird [11]. Dies läßt sich entweder durch aufwendige und komplexe Tests erreichen oder aber, wie in [11] vorgeschlagen, indem kritische Teile der testenden Einheit selbstüberwachend angelegt sind.

SATZ 3.2. (Theorem 3.2. in [11])
Ein Diagnosegraph G=(N,E) ist genau t-diagnostizierbar in Modell BGM, wenn gilt:

1) $d_{in,min} \geq t$
2) Für jedes Knotenpaar $u,v \in N$ mit $d_{in}(u)=d_{in}(v)=t$ und $v \in V(u) \cap N(u)$ gibt es ein $w \in N$, so daß
   entweder $w \in V(u) \setminus V(v)$ und $V(w) \neq V(v)$ gilt
   oder $w \in V(v) \setminus V(u)$ und $V(w) \neq V(u)$.

Ein Strukturalgorithmus für das BGM-Modell

Schritt O: (Initialisierungen)
Der gegebene Graph G sei durch die Adjazenzmatrix A dargestellt ($n=|N|$).
Integervariablen $i,j,k,t,S_r$ $(1 \leq r \leq n)$. Setvariable T.

Schritt 1: Berechne $S_i := \sum_{j=1}^{n} A[j,i]$ $(1 \leq i \leq n)$ und $t := \min_{1 \leq i \leq n} S_i$.
Setze $T := \{i \mid i \in \{1,\ldots,n\}$ und $S_i = t\}$.

Schritt 2: Berechne $V(u_i)$ für alle $i \in T$.

```
FOR i=1 TO n DO
  FOR j=1 TO n DO
    IF i∈T,j∈T,A[i,j]=1,A[j,i]=1 THEN
      BEGIN
          s:=0;
          FOR k=1 TO n DO
            IF (k∈V(u_i),k∉V(u_j),V(u_k)≠V(u_j)) OR
               (k∈V(u_j),k∉V(u_i),V(u_k)≠V(u_i)) THEN
               s:=1;
            IF s=0 THEN Stoppe
               ( G ist (t-1)-diagnostizierbar );
      END;
Stoppe ( G ist t-diagnostizierbar );
```

Falls Mengenoperationen auf den Mengen $V(u_i)$ als Einzelschritte gezählt werden, kommt dieser Algorithmus in $O(n^3)$ Schritten aus.
In [11]wird gezeigt, daß die $D_{1t}$-Graphen t-optimal im BGM-Modell sind.

Ein Diagnosealgorithmus für das BGM-Modell

Schritt 0: (Initialisierungen)
Der Diagnosegraph G mit n Knoten sowie das vorliegende Syndrom seien durch die Matrix A dargestellt, die definiert ist durch

$$A[i,j] = \begin{cases} -1 \text{ falls } (u_i,u_j) \notin E \\ 0 \text{ falls } (u_i,u_j) \in E \text{ und Testergebnis 0} \\ 1 \text{ falls } (u_i,u_j) \in E \text{ und Testergebnis 1} \end{cases}$$

INTEGER t: Grad der Diagnostizierbarkeit von G.

Schritt 1: $\forall u_i \in N$: Existiert ein $u_j \in N$ mit $A[j,i]=1$, so ist $u_i$ intakt. In diesem Falle sind alle $u_j \in N$ mit $A[i,j]=0$ oder $A[j,i]=0$ defekt.

Updaten (siehe unten)
Falls t=0 ist, setze alle noch nicht diagnostizierten Knoten intakt.

Schritt 2: Falls $N \neq \emptyset$ und $t>0$ ist:
Alle $u \in N$ mit $|V(u)| \neq t$ sind defekt.
Updaten (siehe unten)

Schritt 3: Falls $N \neq \emptyset$ und $t>0$ ist:
Berechne $V(u)$ für alle $u \in N$.
Durch Vergleich der Mengen $V(u)$ finde die eindeutige Menge $N'$ von Knoten mit $|N'| = |N| - t$, in der alle Elemente dieselbe Vorgängermenge haben.
Alle $u \in N \setminus N'$ sind defekt, alle $u \in N'$ intakt.

Erläuterung: Unter "Updaten" sind die folgenden Anweisungen verstanden:

$$\begin{cases} N := N \setminus \{\text{diagnostizierte Knoten in diesem Schritt}\}. \\ n := |N| \quad . \\ t := t - |\{\text{diagnostizierte defekte Knoten in diesem Schritt}\}| \, . \\ \text{Alle Einträge, die zu diagnostizierten Knoten in A gehören,} \\ \text{werden gelöscht.} \end{cases}$$

Dieser Diagnosealgorithmus kommt in $O(n^2)$ Schritte aus, wenn eine Mengenoperation in Schritt 3 als ein Schritt gezählt wird. Die Korrektheit des Algorithmus wird analog gezeigt wie die des entsprechenden Algorithmus für Modell VT2 in Abschnitt 4.3.. Deshalb sei hier nicht näher darauf eingegangen.

iii) Das A1-Modell:

Die Modellannahmen von A1 ($t_{01}=X$, $t_{00}=1$) muten nicht sehr realistisch an. Deshalb soll hier nur ein Charakterisierungssatz für Diagnosegraphen bewiesen werden. Als Diagnosealgorithmus kann der für das PMC-Modell entworfene Madden-Algorithmus [50] herangezogen werden.

SATZ 3.3.

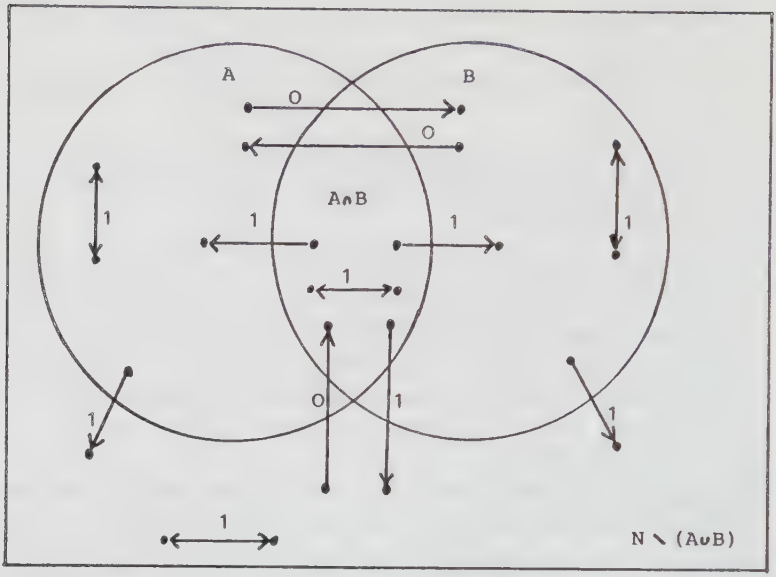
Ein Diagnosegraph $G=(N,E)$ ist genau dann t-diagnostizierbar in Modell A1, wenn gilt:

1) $t \leq (n-1)/2$
2) Für alle $A,B \subseteq N$, $A \neq B$ und $1 \leq |A|, |B| \leq t$ gilt
$$\left[(V(A \setminus B) \not\subseteq B) \vee (V(B \setminus A) \not\subseteq A) \vee (V(A \cap B) \cap (A \cup B) \neq \emptyset)\right] .$$

Beweis:

"$\Longrightarrow$": G sei t-diagnostizierbar in Modell A1. Wäre $t \geq n/2$ und $U \subseteq N$ mit $|U|=t$, so gälte auch $|N \setminus U| \leq t$ und wegen der Modellannahmen könnten U und $N \setminus U$ dasselbe Syndrom hervorrufen (wenn stets $t_{01}=0$ eintritt), im Widerspruch zur t-Diagnostizierbarkeit von G. Also ist $t \leq (n-1)/2$.
Angenommen, Bedingung 2) des Satzes gelte nicht, seien $A,B \subseteq N$ Gegenbeispiele. Dann gilt $V(A \setminus B) \subseteq B$, $V(B \setminus A) \subseteq A$ und $V(A \cap B) \cap (A \cup B) = \emptyset$. Dann können, im Widerspruch zur t-Diagnostizierbarkeit von G, sowohl A als auch B das in Figur 3.4. gezeigte Syndrom erzeugen.

"$\Longleftarrow$": Es gelten die Bedingungen 1) und 2) und G sei nicht t-diagnostizierbar. Dann existieren verschiedene Fehlermuster




Figur 3.4.

$F_1$ und $F_2$ mit nicht mehr als t Elementen, die dasselbe Syndrom hervorrufen können.

Fall a) $F_1 \neq \emptyset$ und $F_2 \neq \emptyset$: Wegen des übereinstimmenden Syndroms muß $V(F_1 \setminus F_2) \subseteq F_2$ und $V(F_2 \setminus F_1) \subseteq F_1$ gelten. Wegen Bedingung 2) gilt dann $V(F_1 \cap F_2) \cap (F_1 \cup F_2) \neq \emptyset$. Ohne Beschränkung der Allgemeinheit existiert also ein Kante von $F_1 \setminus F_2$ nach $F_1 \cap F_2$. Das Testergebnis dieser Kante lautet aber 1 für Fehlermuster $F_1$ und O für $F_2$, so daß dasselbe Syndrom für $F_1$ und $F_2$ nicht möglich ist.

Fall b) Eines der beiden Fehlermuster ist die leere Menge, etwa $F_1 = \emptyset$. Dann gilt $V(F_1 \setminus F_2) = V(\emptyset) = \emptyset \subseteq F_2$ und das von $F_1$ bewirkte Syndrom ist das 1-Syndrom. Außerdem ist $V(F_2 \setminus F_1) = V(F_2) = \emptyset \subseteq F_1$, da sonst bei Auftreten von $F_2$ ein Testergebnis O erschiene. Schließlich gilt $V(F_1 \cap F_2) \cap (F_1 \cup F_2) = V(\emptyset) \cap F_2 = \emptyset$ und Bedingung 2) ist damit verletzt. q.e.d.

iv) Die Modelle S1,S2 und S3:
Die Gemeinsamkeit dieser 3 Modelle besteht darin, daß sie symmetrische Testannahmen machen, d.h. daß stets $t_{ij}=t_{ji}$ $(i,j\in\{0,1\})$ gilt. Die Annahmen der übrigen 6 Modelle sind nicht symmetrisch. Wie das folgende Lemma zeigt, genügt es, in symmetrischen Modellen mit Diagnosegraphen ohne Doppeltests zu arbeiten.

Lemma 3.4.:
Sei Y ein klassisches Modell mit symmetrischen Annahmen, d.h. $Y\in\{S1,S2,S3\}$. Ein Diagnosegraph G ist genau dann t-diagnostizierbar in Modell Y, wenn jede seiner Vereinfachungen G' t-diagnostizierbar in Modell Y ist.

Beweis:
Ist G' t-diagnostizierbar in Modell Y, so sicher auch G (zur Diagnose verwende man nur die in G' auftretenden Kanten). Sei also G t-diagnostizierbar in Modell Y. Die symmetrischen Modellannahmen ermöglichen es, daß die beiden Testergebnisse eines Doppeltests stets übereinstimmen können. Das bedeutet aber, daß sie in diesem Fall nicht mehr Information liefern als ein Einfachtest. q.e.d.

Die Symmetrie der Modellannahmen in den Modellen S1,S2 und S3 bedeutet gerade, daß es für das Fehlerdiagnosevermögen eines Graphen keine Rolle spielt, welche Richtung seine Kanten haben. Da nach Lemma 3.4. der Einbau von Doppeltests in Diagnosegraphen sinnlos ist, stellt damit ein ungerichteter Graph die angemessene Repräsentation einer Diagnosestruktur für diese 3 Modelle dar. Da die Richtung einer Kante keine Bedeutung mehr hat, wird auch die Unterscheidung von testender und getesteter Einheit überflüssig. Zwei Einheiten testen einander, indem sie sich vergleichen. Deshalb ist die Bezeichnung Vergleichstestmodelle für diese Modelle angebracht. In Kapitel 4 werden die ungerichteten Gegenstücke von S1,S2 und S3 ausführlich untersucht. Deshalb seien hier lediglich die 3 Charakterisierungssätze angegeben. Ihre Beweise lassen sich direkt von denen der ungerichteten Modelle in Kapitel 4 übertragen. Nach Lemma 3.4. genügt eine Beschränkung auf Diagnosegraphen ohne Doppeltests.

SATZ 3.4.

Ein Diagnosegraph G=(N,E) ohne Doppeltests ist genau dann t-diagnostizierbar in Modell S1, wenn gilt:

1) $t \leq (n-1)/2 \quad (n=|N|)$
2) $d_{in}(u)+d_{out}(u) \geq t$ für alle $u \in N$
3) $\forall\, 0 \leq p < t \;\forall X \subseteq N \left[ |X| = n-2t+p \Rightarrow |V(X) \cup N(X)| > p \right]$.

SATZ 3.5.

Ein Diagnosegraph G=(N,E) ohne Doppeltests ist genau dann t-diagnostizierbar in Modell S2, wenn gilt:

1) $d_{in}(u)+d_{out}(u) \geq t$ für alle $u \in N$
2) Für alle $u,v \in N$ mit $d_{in}(u)+d_{out}(u)=t=d_{in}(v)+d_{out}(v)$ und $((u,v) \in E \vee (v,u) \in E)$ existiert ein $w \in N$ so, daß
   entweder $w \in C(u) \setminus C(v)$ und $C(w) \neq C(v)$
   oder $w \in C(v) \setminus C(u)$ und $C(w) \neq C(u)$ gilt.
   Dabei ist $C(u) := V(u) \cup N(u)$ zu setzen.

SATZ 3.6.

Ein Diagnosegraph G ohne Doppeltests ist genau dann t-diagnostizierbar in Modell S3, wenn $t \leq (n-1)/2$ gilt.

Man beachte hierbei, daß alle Diagnosegraphen stets als schwach zusammenhängend angenommen werden.

v) Das Modell A2

Modell A2 macht die Annahmen $t_{01}=1$ und $t_{00}=X$. Damit wird also verlangt, daß sich stets das Testergebnis 1 ergibt, falls die getestete Einheit intakt ist - unabhängig von Zustand der testenden Einheit. Eine Realisierung dieser Anforderung läßt sich folgendermaßen denken: In jeder Einheit ist ein Speicherplatz "Testergebnis" auf 1 initialisiert. Ein Änderung dieses Wertes wird nur durch ein fehlerhaftes Verhalten der von dieser Einheit getesteten Einheit zugelassen.

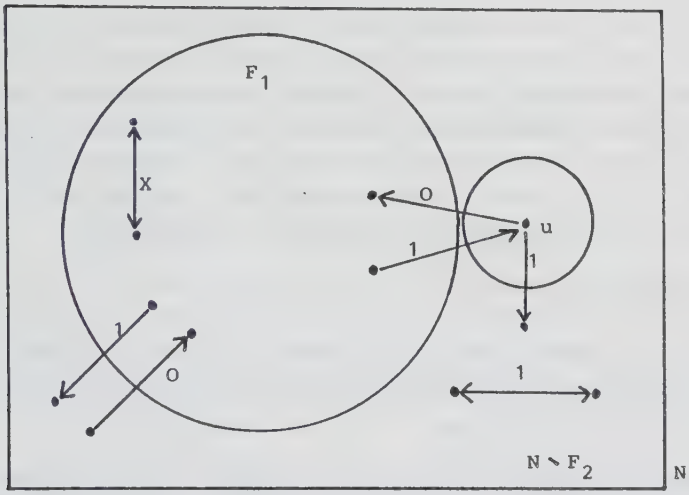
SATZ 3.7.

Ein Diagnosegraph G=(N,E) ist genau dann t-diagnostizierbar in Modell A2, wenn $d_{in,min} \geq t$ gilt.

Beweis:

1) Sei G t-diagnostizierbar in Modell A2 und sei $d_{in}(u) \leq t-1$ für

ein $u \in N$. Setze $F_1 := V(u)$ und $F_2 := F_1 \cup \{u\}$. Dann enthalten $F_1$ und $F_2$ jeweils höchstens t Elemente und können beide das in Figur 3.5. eingezeichnete Syndrom erzeugen, im Widerspruch zur t-Diagnostizierbarkeit von G.




Figur 3.5.

2) Es gilt $d_{in,min} \geq t$. Bei Vorliegen von höchstens t defekten Einheiten gelingt dann die eindeutige Diagnose jeder Einheit folgendermaßen: $u \in N$ ist intakt, falls alle Tests über u das Testergebnis 1 ergeben; $u \in N$ ist defekt, falls ein Test über u das Ergebnis 0 ergibt.
Man beachte dabei, daß wegen $d_{in}(u) \geq t$ in der Menge V(u) einer defekten Einheit mindestens eine intakte Einheit enthalten sein muß. q.e.d.

Der höchste erreichbare Grad t der Diagnostizierbarkeit ist in Modell A2 für einen gegebenen Graphen G sehr leicht festzustellen. Der Wert $t = d_{in,min}$ leistet das gewünschte.
Teil 2) des Beweises von Satz 3.7. stellt gleichzeitig auch die Beschreibung eines Diagnosealgorithmus für Modell A2 dar, der

eine Laufzeit von $O(n^2)$ bzw. $O(|E|)$ je nach verwendeter Datenstruktur des Graphen (Adjazenzmatrix oder Adjazenzliste) hat. Die $D_{1t}$-Graphen erfüllen die Bedingung $d_{in}(u)=t$ für alle $u \in N$ und sind deshalb t-optimale Diagnosegraphen für Modell A2.

vi) Das Modell A3:

Modell A3 macht die Annahme, daß ein Testergebnis stets den korrekten Zustand der getesteten Einheit verrät. Das setzt voraus, daß diejenigen Teile einer Einheit, die für die Durchführung eines Tests einer anderen Einheit gebraucht werden, zur Hardcore der Einheit gehören, also völlig fehlersicher sind. Eine Erfüllung dieser starken Anforderung bringt, wie Satz 3.8. zeigt, dafür die Notwendigkeit nur einfacher Diagnosestrukturen, um hohe Grade der Diagnostizierbarkeit zu erreichen.

SATZ 3.8.

Sei $G=(N,E)$ ein Diagnosegraph mit $|N|=n$ und sei $1 \leq t \leq n$. Dann gilt: G ist t-diagnostizierbar in Modell A3 genau dann, wenn G n--diagnostizierbar in Modell A3 ist. Beide Bedingungen sind äquivalent zu $d_{in,min} \geq 1$.

Beweis: Die eindeutige Diagnose einer Einheit gelingt stets, wenn diese von mindestens einer anderen Einheit getestet wird.

q.e.d.

Diagnosegraphen in Modell A3 haben also immer entweder die volle n-Diagnostizierbarkeit oder keinerlei Fehlerlokalisierungsvermögen. Der Testring $D_{11}$ mit n Knoten erfüllt die Bedingung $d_{in}(u)=1$ für alle $u \in N$ und ist deshalb n-optimal in Modell A3. Eine Überprüfung der Bedingung $d_{in,min} \geq 1$ gelingt in n Schritten, ein Diagnosealgorithmus von n Schritten ergibt sich sofort aus den Modellannahmen. Man betrachte einen Test über jede Einheit und nehme das Testergebnis als Zustand der Einheit.

vii) Das Modell A4:

In einer Realisierung von Modell A4 darf nur eine intakte Einheit in der Lage sein, das Testergebnis O hervorzubringen (beim Test einer defekten Einheit). Etwa kann ein Speicherplatz "Testergebnis" auf 1 initialisiert sein und eine Änderung dieses Wertes kann die Einheit nur nach Durchlaufen eines Selbsttests, den im allgemeinen nur eine intakte Einheit besteht, vornehmen.

SATZ 3.9.

Ein Diagnosegraph G=(N,E) ist t-diagnostizierbar in Modell A4 genau dann, wenn gilt:

1) $t \leq n-1$ $(n=|N|)$ .

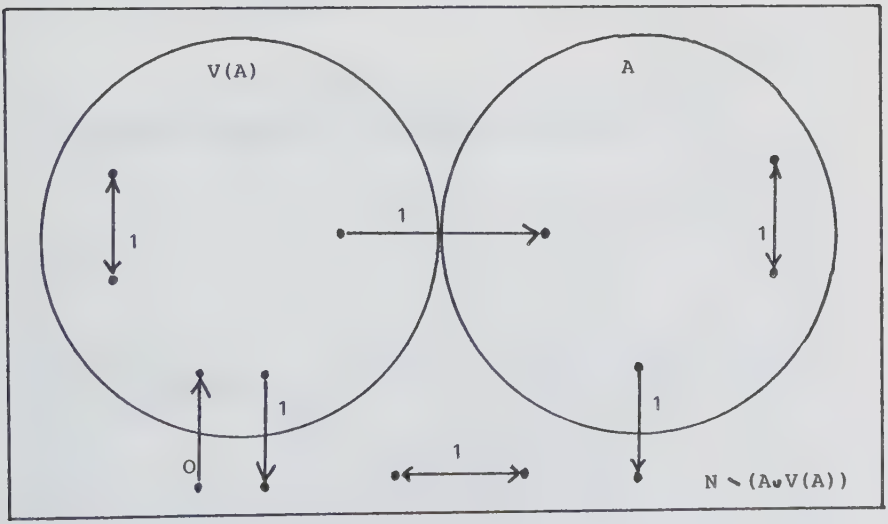
2) $\forall A \in DTK(G): |V(A)| > t-|A|$ .

Beweis:

1) Sei G t-diagnostizierbar in Modell A4. Der Wert t=n ist nicht erreichbar, da sonst die beiden Fehlermuster $F_1=N$ und $F_2=\emptyset$, beide mit nicht mehr als t Elementen, das 1-Syndrom hervorbringen würden. Angenommen, Bedingung 2) gelte nicht und $A \in DTK(G)$ sei ein Gegenbeispiel (man beachte $A \neq \emptyset$ und $|A| \leq t$). Setzt man dann $F_1 := V(A)$ und $F_2 := V(A) \cup A$, so gilt

$$|F_1| = |V(A)| \leq t-|A| < t \text{ und } |F_2| = |V(A)| + |A| \leq t-|A|+|A| = t.$$

Figur 3.6. zeigt, daß $F_1$ und $F_2$ dasselbe Syndrom hervorrufen.




Figur 3.6.

Man beachte dabei, daß $V(A) \cap N(A) = \emptyset$ ist, da A eine vollständige Doppeltestkomponente ist.

2) Es gelten die Bedingungen 1) und 2) aus Satz 3.9.. Sicher ist $\bigcup_{A \in DTK(G)} A = N$, sei nun $A \in DTK(G)$.

Fall a) $|A|=1$, etwa $A=\{u\}$. Nach Bedingung 2) gilt dann $d_{in}(u) > t-1$, also $|V(u)| \geq t$ Tests existieren über u. Dann ist u genau dann defekt, wenn mindestens einer dieser Tests das Ergebnis O ergibt.
Fall b) $|A|>1$: Falls alle Tests innerhalb von A das Ergebnis 1 haben, besteht A aus lauter intakten Einheiten, wenn kein Test von V(A) über ein Element von A den Wert O ergibt, und aus lauter defekten Einheiten sonst (man beachte Bedingung 2)). Ergab die Testkante $(u,v) \in E$ (dabei $u,v \in A$) das Ergebnis O, so muß die dann existierende Kante (v,u) das Ergebnis 1 bringen (nach den Modellannahmen). Dann ist u intakt und v defekt. Ausgehend von u und v lassen sich nun alle Elemente in A eindeutig diagnostizieren: Sei $w_1 \in A$ schon diagnostiziert und $\{(w_1,w_2),(w_2,w_1)\} \subseteq E$ (also $w_2 \in A$). Sind beide Kanten mit 1 bewertet, bekommt $w_2$ denselben Zustand wie $w_1$ zugewiesen. Ist $(w_1,w_2)$ mit O bewertet, ist $w_2$ defekt; ist $(w_2,w_1)$ mit O bewertet, ist $w_2$ intakt. Damit sind alle $u \in A$ eindeutig diagnostizierbar und G damit t-diagnostizierbar. q.e.d.

Korollar 3.1.:
Ist G ein Diagnosegraph ohne Doppeltests, so gilt:
G ist t-diagnostizierbar in Modell A4 $\Longleftrightarrow d_{in,min} \geq t$.

Korollar 3.2.:
Jeder Diagnosegraph G=(N,E) ohne Einzeltests ist $(|N|-1)$-diagnostizierbar in Modell A4.

Beide Korollare folgen sofort aus Satz 3.9..

Ein Strukturalgorithmus für Modell A4

Schritt 1: Bestimme alle $A \in DTK(G)$ und deren Vorgängermengen V(A) mit Hilfe der Tiefensuche ("depth-first-search" [71]).
Schritt 2: Setze $t := \min\{|V(A)| + |A| \mid A \in DTK(G)\} - 1$.
G ist t-diagnostizierbar.

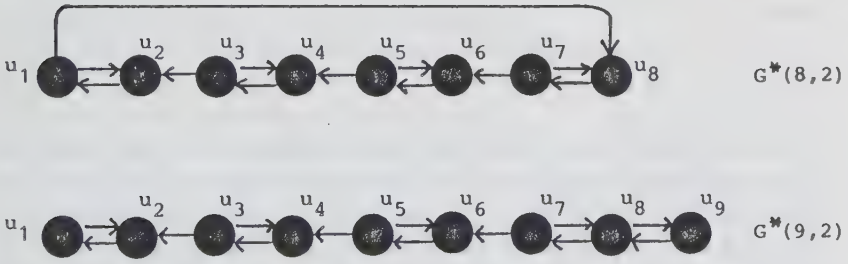
Dieser Algorithmus berechnet mit t den höchsten möglichen Diagnostizierbarkeitsgrad in $O(|N|^2)$ Schritten (bei Verwendung einer Adjazenzmatrix für G).

Ein Diagnosealgorithmus wurde im zweiten Teil des Beweises von Satz 3.9. schon angegeben, auch dieser kommt in $O(|N|^2)$ Schritten aus.

Definition:
Sei $n\in\mathbb{N}$, $n\geq 3$. Der Graph $G^*(n,t)=(N^*,E^*)$ mit $|N^*|=n$ wird definiert durch

$$E^*:=\{(u_i,u_{i-1})\mid i=2,3,..,n\}\cup\{(u_i,u_{i+1})\mid i\in\{1,2,..,n-1\}\setminus\{st\mid s=1,2,..,k-1\}\}\quad (k=\lfloor\tfrac{n}{t}\rfloor)$$

Falls $kt=n$ gilt, wird zu $E^*$ die Kante $(u_1,u_n)$ hinzugefügt.
Man beachte, daß dann $G^*(n,1)=D_{11}$ gilt; Figur 3.7. zeigt weitere Beispiele.




Figur 3.7. Beispiele von $G^*(n,t)$-Graphen.

Proposition 3.1.:
Der Diagnosegraph $G^*:=G^*(n,t)$ ist t-optimal in Modell A4 für $1\leq t\leq n-1$.

Beweis:
1) G ist t-diagnostizierbar in Modell A4:
G besteht aus $k=\lfloor n/t\rfloor$ Doppeltestkomponenten A, alle mit $|A|\geq t$. Für $A\in DTK(G)$ mit $|A|=t$ gilt stets $|V(A)|=1$ und damit $|V(A)|>t-|A|$; für $|A|>t$ ist $V(A)=\emptyset$. Damit ist Bedingung 2) aus Satz 3.9. erfüllt.
2) Die t-Optimalität von G :
Es gilt $|E^*|=2(n-1)-(k-1)+\alpha$ mit $\alpha=0$ für $kt<n$ und $\alpha=1$ für $kt=n$.

a) Sei nun $G=(N,E)$ ein t-optimaler Graph in Modell A4, zu zeigen ist dann $|E|=|E^*|$. Sicher ist $\bigcup_{A\in DTK(G)} A=N$, sei $A\in DTK(G)$. Falls $|A|\leq t$ gilt, existieren innerhalb von A mindestens $2(|A|-1)$ Tests, wegen Satz 3.9. sind außerdem noch mindestens $t-|A|+1$ Tests von außerhalb notwendig. Insgesamt sind also mindestens $|A|+t-1$ Tests für die Einheiten von A vorgesehen. Gilt $|A|>t$, so werden innerhalb von A wenigstens $2(|A|-1)$ Tests gebraucht.

b) Gäbe es $A_1, A_2 \in DTK(G)$ mit $|A_1|<t$, so ergäbe die Zusammenlegung zu einem Doppeltestbaum $A_3$ mit $A_3=A_1 \cup A_2$ keine Kantenerhöhung. Für $|A_2|<t$ gilt nämlich

$$\max(2(|A_1|+|A_2|-1), |A_1|+|A_2|+t-1) \leq |A_1|+|A_2|+2t-2,$$

für $|A_2|\geq t$

$$2(|A_1|+|A_2|-1) \leq |A_1|+t-1+2(|A_2|-1) = |A_1|+2|A_2|+t-1.$$

Ohne Einschränkung kann also $|A|\geq t$ für alle $A\in DTK(G)$ angenommen werden.

c) Für $t>n/2$ besteht dann G aus genau einem $A\in DTK(G)$ mit $|A|=n$ und es gilt $|E|=2(n-1)=|E^*|$.

d) Sei nun $t\leq n/2$:

G ist zusammenhängend und t-optimal in Modell A4, die verschiedenen $A\in DTK(G)$ müssen also durch Einzeltests miteinander verbunden sein. Für $|A|=t$ muß wegen Bedingung 2) aus Satz 3.9. sogar ein Einzeltest $(u,v)$ mit $u\in V(A)$ und $v\in A$ existieren.

Eine Erhöhung von $|DTK(G)|$ wirkt kantenzahlvermindernd, da diese $A\in DTK(G)$ nur durch jeweils einen Einzeltest verbunden sein müssen. G besteht deshalb aus $k=\lfloor n/t \rfloor$ Doppeltestkomponenten, die alle Doppeltestbäume sind und einen Baum von Doppeltestkomponenten bilden. Für $kt=n$ werden k verbindende Einzeltests gebraucht. Also gilt $|E|=2(n-1)-(k-1)+a=|E^*|$. q.e.d.

## 3.3. UNTERSUCHUNG DER KLASSISCHEN MODELLE BEZÜGLICH T-DETEKTIERBARKEIT

Für ein System bzw. für einen Diagnosegraphen bedeutet t-Detektierbarkeit, daß bei Auftreten von höchstens t defekten Einheiten aus der Kenntnis des Syndroms heraus entschieden werden kann, ob defekte Einheiten vorliegen. Dies bedeutet, daß bei t´ defekten Einheiten ($1 \leq t' \leq t$) mindestens ein Testbit des Syndroms den Wert O annehmen muß, da das 1-Syndrom stets mit dem völlig intakten System in Verbindung gebracht wird.

Für die graphentheoretische Charakterisierung der t-Detektierbarkeit spielt in einigen der klassischen Modelle der sogenannte "Abgeschlossenheitsindex" c(G) (closure index) eines Graphen G=(N,E) die entscheidende Rolle. Dieser ist nach [75] als $c(G) := \min(\{|U| \mid \emptyset \neq U \subseteq N,\ V(U) = \emptyset\})$ definiert.

SATZ 3.10.

Sei G=(N,E) ein Diagnosegraph mit n=|N|.

a) Sei Y eines der Modelle PMC, A1, A2 oder A4 und sei $t \leq n-1$. Dann ist G t-detektierbar in Modell Y genau dann, wenn $c(G) \geq t+1$ gilt.

b) Sei Modell Y eines der Modelle BGM oder A3. Dann ist G n-detektierbar in Modell Y genau dann, wenn $d_{in,min} \geq 1$ gilt.

c) Sei Modell Y eines der Modelle S1 oder S3. Dann ist jeder Diagnosegraph (n-1)-detektierbar in Modell Y.

d) Jeder Diagnosegraph ist n-detektierbar in Modell S2.

Beweis:

a) Gilt $c(G) \geq t+1$ und ist $F \neq \emptyset$ ein Fehlermuster von höchstens t Elementen, so gilt $V(F) \neq \emptyset$, d.h. mindestens ein $u \in F$ wird von einer intakten Einheit getestet; dies ergibt ein O-Testbit. Also ist G t-detektierbar in Modell Y.
Ist hingegen $c(G) \leq t$, so existiert ein $F \subseteq N$ mit $|F| \leq t$ und $V(F) = \emptyset$. Modell Y läßt $t_{01}=1$ und $t_{00}=1$ zu. Also kann durch F das 1-Syndrom hervorgerufen werden und G ist nicht t-detektierbar.

b) Ist $d_{in,min} \geq 1$, so wird jede Einheit getestet. Ein Test einer defekten Einheit ergibt in Modell Y stets das Ergebnis O.

Existiert dagegen ein $u \in N$ mit $d_{in}(u)=0$, so kann das Fehlermuster $F=\{u\}$ das 1-Syndrom bewirken. G ist dann nicht einmal 1-detektierbar.

c) Sei F ein Fehlermuster mit $|F| \leq n-1$ und sei $u \in N$ eine (existierende) intakte Einheit. Sei $v \in F$. Wegen des Zusammenhangs von G existiert ein Weg von u nach v, d.h. eine endliche Folge $w_1, w_2, \ldots, w_k$ $(w_i \in N)$ mit $w_1=u$, $w_k=v$ und $(w_i, w_{i+1}) \in E \vee (w_{i+1}, w_i) \in E$ für alle $1 \leq i \leq k-1$. Sei j die kleinste Zahl mit $2 \leq j \leq k$ so, daß $w_j$ eine defekte Einheit ist. Dann ist $w_{j-1}$ eine intakte Einheit und $(w_{j-1}, w_j) \in E \vee (w_j, w_{j-1}) \in E$. Ein Test zwischen einer intakten und einer defekten Einheit (in welcher Richtung auch ausgeführt) ergibt stets das Ergebnis 0 wegen der symmetrischen Annahmen von Modell Y.

d) Nach den Annahmen von Modell S2 ($t_{01}=t_{10}=t_{00}=0$) ergibt jeder Test $(u,v) \in E$ das Ergebnis 0, wenn wenigstens eine der Einheiten u oder v defekt ist. q.e.d.

Man beachte, daß für die beiden Modelle S1 und S3 in Teil c) von Satz 3.10. niemals n-Detektierbarkeit möglich ist, da $t_{00}=1$ zugelassen ist.

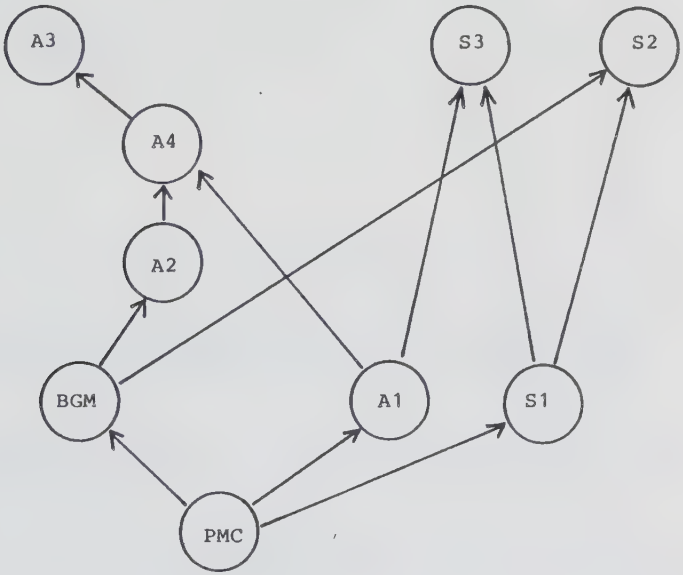
Die Bestimmung des höchsten möglichen Grads der t-Detektierbarkeit macht nunmehr nur für die 4 Modelle in Teil a) des Satzes einen nennenswerten Rechenaufwand notwendig. Zur Bestimmung von c(G) sei auf [75] verwiesen. Die Aussage für das PMC-Modell ist ein Spezialfall von Theorem 4 in [75].

## 3.4. BEZIEHUNGEN ZWISCHEN DEN MODELLEN

In den beiden vorausgegangenen Abschnitten wurden die 9 klassischen Modelle separat untersucht. Hier soll nun nach logischen Beziehungen zwischen ihnen gefragt werden. Zu diesem Zweck wird eine Implikationsrelation $\longrightarrow$ auf der 9-elementigen Menge der klassischen Modelle eingeführt:

Seien $Y_i$ klassische Modelle (i=1,2). Dann bestehe die Relation $\longrightarrow$ zwischen $Y_1$ und $Y_2$ (in Zeichen $Y_1 \longrightarrow Y_2$), falls gilt: Für alle Diagnosegraphen G folgt aus der t-Diagnostizierbarkeit von G in Modell $Y_1$ auch die t-Diagnostizierbarkeit von G in Modell $Y_2$. Diese Relation stellt eine Ordnungsrelation auf der Menge M der klassischen Modelle der Selbstdiagnose dar (die Antisymmetrie

läßt sich sofort mit den Charakterisierungssätzen 3.1.-3.9. nachprüfen). $(M, \longrightarrow)$ bildet sogar einen Halbverband, d.h. je zwei Modelle $Y_1, Y_2 \in M$ besitzen eine größte untere Schranke $Z \in M$ (also $Z \longrightarrow Y_1$, $Z \longrightarrow Y_2$ und, falls $Z'$ ebenfalls diese Eigenschaft hat, so $Z' \longrightarrow Z$). Figur 3.8. zeigt die Struktur dieses Halbverbandes.




Figur 3.8. Der Halbverband der klassischen Modelle bezüglich t-Diagnostizierbarkeit.

Wegen der Transitivität der Implikationsrelation gilt dann natürlich auch etwa PMC $\longrightarrow$ S3 oder A2 $\longrightarrow$ A3.
Die Richtigkeit der in Figur 3.8. gezeigten Struktur von $(M, \longrightarrow)$ läßt sich mit den Sätzen 3.1.-3.9. nachprüfen, ebenso auch das Nichtbestehen der Relation zwischen Modellen, die durch eine transitive Vervollständigung von Figur 3.8. nicht verbindbar sind (also A2 $\not\longrightarrow$ S3).

Das Bestehen der Relation $Y_1 \longrightarrow Y_2$ zwischen zwei Modellen läßt sich folgendermaßen interpretieren. $Y_1$ macht die schwächeren Modellannahmen und stellt deshalb höhere Anforderungen an einen Diagnosegraph, um einen bestimmten Grad t der Diagnostizierbarkeit zu erreichen. Dies ist in Fällen, in denen etwa in $Y_1$ $t_{O1}=X$ und in $Y_2$ $t_{O1}=1$ oder O gilt, sofort einsichtig. Jedoch gilt auch BGM $\longrightarrow$ A2 und hier gilt zwar $t_{O1}=X$ in BGM und $t_{O1}=1$ in A2, aber auch $t_{OO}=O$ in BGM und $t_{OO}=X$ in A2.

Analoge Überlegungen kann man bezüglich der Detektierbarkeit machen. Abschnitt 3.3. zeigt, daß im wesentlichen 4 Typen von klassischen Modellen bezüglich t-Detektierbarkeit bestehen:

Typ 1 = {PMC,A1,A2,A4}.

Typ 2 = {BGM,A3}.

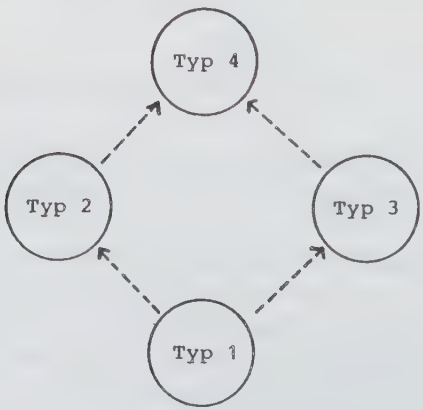
Typ 3 = {S1,S3}.

Typ 4 = {S2}.

Sei nun M´:={Typ 1,Typ 2,Typ 3,Typ 4} und sei die Implikationsrelation ---→definiert durch:

Typ i ---→Typ j , falls aus der t-Detektierbarkeit eines Diagnosegraphen G in einem der Modelle aus Typ i auch die t-Detektierbarkeit von G in jedem der Modelle aus Typ j folgt ($1 \leq i,j \leq 4$).

Die Struktur des Verbandes (M´,---→) zeigt Figur 3.9.




Figur 3.9. Der Verband der Modelltypen bezüglich Detektierbarkeit

Zum Beweis der Richtigkeit von Figur 3.9. beachte man, daß aus der t-Detektierbarkeit in einem Modell Y∈Typ 1 stets $t \leq n-1$ und $d_{in,min} \geq 1$ (sonst $c(G)=1$ und $t=0$) folgt.

Anmerkung:

In [41] werden ebenfalls die Beziehungen zwischen den 9 Modellen untersucht. Das Bezugsmaß für die Güte von Diagnosestrukturen ist dabei aber die s-t-Diagnostizierbarkeit (vgl. Kapitel 2), als Diagnosestrukturen werden nur die $D_{11}$-Graphen betrachtet. Unter diesen Voraussetzungen ergibt sich dann eine Verbandsstruktur.

# 4. STATISCHE VERGLEICHSTESTMODELLE

In Kapitel 3 wurden Modelle beschrieben, die gerichtete Tests verwenden, d.h. zwischen testenden und getesteten Einheiten unterscheiden. Doch wie kann festgelegt werden, ob die Signale, die von getesteten zu testenden Einheiten gelangen, korrekt sind ? Die Tester müssen offensichtlich diese Signale mit ihnen zur Verfügung stehenden oder von ihnen ebenfalls berechneten Referenzsignalen vergleichen.

In diesem Kapitel nun sollen sogenannte Vergleichstestmodelle vorgestellt werden. Ein Test besteht hier aus einem Vergleich zweier Einheiten bzw. der Daten, die beide Einheiten berechnen. Bis auf die Verwendung von ungerichteten Tests werden dieselben Parameter für Selbstdiagnosemodelle verwendet wie bei den klassischen Modellen in Kapitel 3.

Vor allem beim Einsatz in homogenen Multiprozessorsystemen haben Vergleichstests gegenüber gerichteten Tests den Vorteil, daß sie

- einfach zu implementieren sind
- nicht hardwareabhängig sind
- keine spezialisierte Hardware zur Testauswertung brauchen.

Abschnitt 4.1. stellt die drei zu behandelnden Vergleichstestmodelle vor und stellt die eindeutige Beziehung zu den drei klassischen Modellen S1, S2 und S3 aus Abschnitt 3.2. her. In der Abschnitten 4.2. - 4.4. werden diese drei Modelle untersucht, entsprechend [5]. Abschnitt 4.5. schließlich interpretiert die statische Selbstdiagnose mit Vergleichstestmodellen als Lösung eines Suchproblems ohne Gedächtnis und macht weitere Anmerkungen zu den Vergleichstestmodellen.

## 4.1. BEZEICHNUNGEN UND MODELLANNAHMEN

Für ein System S mit n Einheiten $N=\{u_1,u_2,\ldots,u_n\}$ $(n\geq 3)$ läßt sich die Diagnosestruktur hier durch einen endlichen, ungerichteten Diagnosegraphen $G=(N,E)$ beschreiben, wobei $E:=\{(u_i,u_j)\,|\,u_i,u_j\in N, u_i$ und $u_j$ vergleichen einander$\}$ ist. Ein Diagnosegraph besitzt keine Schlingen $(u_i,u_i)$ und keine Mehrfachkanten. Die Begriffe Syndrom, Fehlermuster und konsistentes Fehlermuster werden analog zu 3.1. definiert. Das O-Syndrom ist das Syndrom mit lauter O-Ergebnissen. Es ist vernünftig anzunehmen, daß der Vergleich zweier intakter Einheiten stets Übereinstimmung $t_{11}=1$ ergibt und der Vergleich einer intakten und einer defekten Einheit niemals Übereinstimmung $t_{01}=0$ (hier bedeutet $t_{ij}$ $(i,j\in\{0,1\})$ den Testausgang eines Vergleichstests, bei dem sich eine Einheit im Zustand i und eine Einheit im Zustand j befindet ).Die Annahmen $t_{11}=1$ und $t_{01}=0$ bedeuten Vollständigkeit der Tests. Aus der Menge $\{0,1,X\}$ frei zu wählen bleibt somit nur $t_{00}$ (dabei wieder $t_{00}=X \leftrightarrow$ beide Testausgänge sind zugelassen). Hier die Bezeichnungen der drei möglichen Modelle mit ihren Annahmen (Figur 4.1.):

| | | $t_{ij}$ | | |
|---|---|---|---|---|
| i | j | Modell VT1 | Modell VT2 | Modell VT3 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | | | |
| 0 | 0 | X | 0 | 1 |

Figur 4.1. Die 3 möglichen Vergleichstestmodelle

Für einen (ungerichteten) Diagnosegraphen $G=(N,E)$ seien definiert [36]:

$N(u) := \{u_i \mid u_i \in N, (u_i,u) \in E\}$ Nachbarmenge von $u \in N$.

$d(u) := |N(u)|$ Grad von $u \in N$, $d_{min} := \min_{u \in N} d(u)$ Minimalgrad von $G$.

$N(U) := \bigcup_{u \in U} N(u) \setminus U$ (echte) Nachbarmenge von $U \subseteq N$.

Definition [36]: Seien $G_i=(N_i,E_i)$ $(i=1,2)$ ungerichtete Graphen. Ihre Vereinigung $G_u=(N_u,E_u)$ (in Zeichen $G_u=G_1 \cup G_2$) ist durch $N_u := N_1 \cup N_2$ und $E_u := E_1 \cup E_2$ definiert, ihr Verbund $G_v=(N_v,E_v)$ (in Zeichen $G_v=G_1+G_2$) durch $N_v := N_1 \cup N_2$ und $E_v := E_1 \cup E_2 \cup \{(u,v) \mid u \in N_1, v \in N_2\}$.

Alle benutzten Graphen seien zusammenhängend [36], die t-Diagnostizierbarkeit und t-Optimalität von Diagnosegraphen sei analog zu 3.2. definiert.

Die 3 Vergleichstestmodelle entsprechen umkehrbar eindeutig den 3 klassischen Modellen S1, S2 und S3 mit symmetrischen Modellannahmen:

> Ist G ein gerichteter, t-diagnostizierbarer Graph für Modell Si ohne Doppeltests, so ist $G_{ung}$ (der ungerichtete Graph, der aus G durch Ersetzen der gerichteten durch ungerichtete Kanten entsteht) t-diagnostizierbar in Modell VTi (i=1,2,3) und umgekehrt. (Beim Übergang von einem ungerichteten zum gerichteten Graphen kann dabei jeweils eine der beiden Kantenrichtungen beliebig gewählt werden)

Anmerkung:

Man beachte bei den Definitionen in diesem Abschnitt, daß teilweise dieselben Symbole für analoge Sachverhalte bei gerichteten und ungerichteten Graphen verwendet werden. Aus dem Zusammenhang wird aber jeweils eindeutig hervorgehen, ob gerade der gerichtete oder ungerichtete Fall vorliegt.

## 4.2. DAS VERGLEICHSTESTMODELL VT1

Modell VT1 macht von allen 3 Vergleichstestmodellen die schwächsten Modellannahmen, da es daß Testergebnis $t_{OO}$ nicht festlegt. Bedingt durch diese Sachlage, kann das Modell stets eingesetzt werden, wenn die Teststrategie von Vergleichen zweier Einheiten benutzt wird. Die beiden anderen Modelle VT2 und VT3 müssen auf begrenzte Klassen von Anwendungsfällen beschränkt bleiben. Der folgende Struktursatz für Modell VT1 ist in Analogie zu Theorem 2 in [35] formuliert.

SATZ 4.1.

Ein Diagnosegraph G=(N,E) ist t-diagnostizierbar in Modell VT1 genau dann, wenn gilt

1) $t \leq (|N|-1)/2$
2) $d_{min} \geq t$
3) $\forall\, 0 \leq p < t \;\; \forall\, U \subseteq N \;\; \left[ |U| = |N| - 2t + p \Longrightarrow |N(U)| > p \right]$ .

Beweis:

Sei G t-diagnostizierbar. Wäre $t \geq |N|/2$ und etwa $F \subseteq N$ mit $|F| = t$, so könnten durch F und $N \setminus F$ dasselbe Syndrom erzeugt werden. Wäre $d_{min} < t$, etwa $d(u) < t$ für ein $u \in N$, so bewirkten N(u) und $N(u) \cup \{u\}$ dasselbe Syndrom und hätten beide höchstens t Elemente (analog zu [69]).

Die Notwendigkeit von Bedingung 3) sowie die Notwendigkeit der t-Diagnostizierbarkeit von G bei Gültigkeit von 1)-3) werden analog zum Beweis von Theorem 2 in [35] gezeigt. Statt der dortigen Nachfolgermenge U wähle man die Nachbarmenge N(U), gerichtete Kanten ersetze man durch ungerichtete Kanten. q.e.d.

Ein Strukturalgorithmus für Modell VT1, der in polynomialer Zeit arbeitet, ist nicht bekannt. Hier zwei hinreichende Bedingungen, deren Gültigkeit effektiv nachgeprüft werden kann.

Proposition 4.1.

Sei G=(N,E) ein (ungerichteter) Graph und $t \leq \min\{(|N|-1)/2, K(G)\}$ (K(G) Knotenzusammenhangsgrad von G [36]). Dann ist G t-diagnostizierbar in Modell VT1.

Beweis: analog zu [35].

Proposition 4.2.:
Sei G ein (ungerichteter) Graph und sei $t := \lfloor d_{min}/2 \rfloor$.
Dann ist G t-diagnostizierbar in Modell VT1.

Beweis: Bei Auftreten von höchstens t defekten Knoten gelingt die eindeutige Diagnose für jedes $u \in N$: u ist genau dann intakt, wenn sich im Vergleich mit mindestens t seiner wenigstens 2t Nachbarknoten Übereinstimmung ergibt. q.e.d.

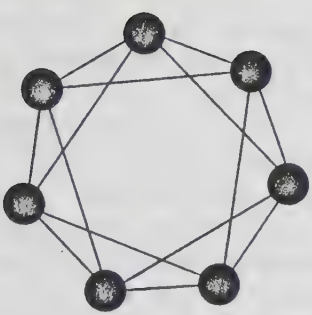
Ein effektiver Algorithmus für die Frage, ob der (Knoten-) Zusammenhangsgrad eines Graphen mindestens k ist, wird in [29] angegeben. Für Werte $k < \sqrt{|N|}$ genügen $O(k \cdot |N|^3)$ Schritte, im allgemeinen Fall $O(k^3 \cdot |N|^2 + k \cdot |N|^3)$ Schritte. Die hinreichende Bedingung der zweiten Proposition ist in $O(|N|^2)$ Schritten nachzuprüfen (bei Verwendung einer Adjazenzmatrix).

Das nächste Ziel der Untersuchung von Modell VT1 ist die Angabe von t-optimalen Diagnosegraphen mit n Knoten für alle möglichen Parameter $n \geq 3$ und $1 \leq t \leq (n-1)/2$.

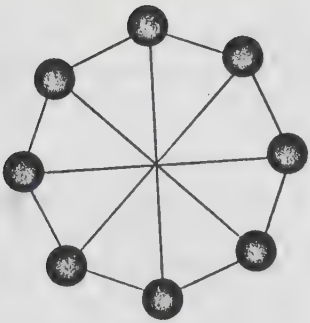
Definition: Sei $n \geq 3$ und $1 \leq k \leq \lfloor (n-1)/2 \rfloor$. Folgende Graphen mit Knotenmenge $\{u_1, u_2, \ldots, u_n\}$ werden definiert:

1) Der Graph $D_{1k}(n)$: $u_i, u_j$ adjazent $\Leftrightarrow j \in \{i-k, .., i-1, i+1, .., i+k\}$ (Addition von Indizes modulo n).
2) Der Graph D(n) für gerades n:
   $u_i, u_j$ adjazent $\Leftrightarrow |i-j| = n/2$.
3) Der Graph S(n) für ungerades n:
   $u_i, u_j$ adjazent $(i<j) \Leftrightarrow 1 \leq i \leq \lceil n/2 \rceil$ und $j = i + \lfloor n/2$
4) Die Testkette TK(n): $u_i, u_j$ adjazent $\Leftrightarrow |i-j| = 1$.

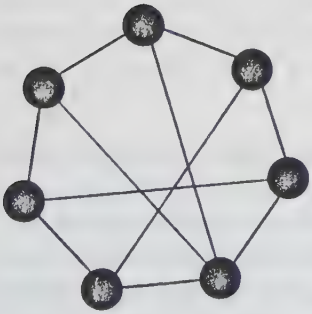
Die Graphen $D_{1k}(n)$, D(n) und S(n) wurden unabhängig auch in [53] eingeführt. Beispiele dieser Definitionen werden in Figur 4.2. gezeigt.



$D_{12}(7)$



$D_{11}(8) \cup D(8)$



$D_{11}(7) \cup S(7)$

$TK(5)$

Figur 4.2. Beispiele von Diagnosegraphen

Proposition 4.3.:
Eine Liste t-optimaler Diagnosegraphen $G=(N,E)$ in Modell VT1 für alle zulässigen Parameter $n=|N| \geq 3$ und $1 \leq t \leq (n-1)/2$ ist gegeben durch Figur 4.3.

Beweis:
1) $TK(n)$ ist 1-optimal: Die Kantenzahl n-1 ist minimal für einen zusammenhängenden Graphen mit n Knoten. $TK(n)$ ist 1-diagnostizierbar in VT1, da $1 \leq (n-1)/2$ und $K(TK(n))=1$ gilt (Proposition 4.1.)
2) Für $t>1$ haben alle angegebenen Graphen die für t-diagnostizierbare Graphen minimale Kantenzahl $\lceil nt/2 \rceil$ (Satz 4.1. schreibt $d_{min} \geq t$ und deshalb $|E| \geq nt/2$ für t-diagnostizierbare Graphen vor).
3) Alle angegebenen Graphen sind t-diagnostizierbar in Modell VT1.

| Parameter | Diagnosegraph |
|---|---|
| t=1 | TK(n) |
| t gerade | $D_{1k}(n)$ mit k=t/2 |
| t>1 ungerade und n gerade | $D_{1k}(n) \cup D(n)$ mit k=(t-1)/2 |
| t>1 ungerade und n ungerade | $D_{1k}(n) \cup S(n)$ mit k=(t-1)/2 |
| ( dabei haben $D_{1k}(n)$ und D(n) bzw. $D_{1k}(n)$ und S(n) dieselben Knotenmengen ) | |

Figur 4.3.

Der Beweis gelingt nach Proposition 4.1. durch die Nachprüfung von $K(G) \geq t$. Dies sei exemplarisch für den Fall $G=D_{1k}(n) \cup D(n)$ (also t ungerade, t=2k+1 und n gerade) durchgeführt.
Ein Zusammenhangsgrad $K(G) \geq t$ bedeutet, daß die Entfernung beliebiger t-1 Knoten (und der inzidenten Kanten) einen zusammenhängenden Restgraphen übrigläßt.
Seien also aus G t-1=2k Knoten entfernt und sei V die Menge der entfernten Knoten.
Fall a) $V=\{u_i,u_{i+1},\ldots,u_{i+k}\} \cup \{u_j,u_{j+1},\ldots,u_{j+k}\}$ (Indizes mod n) mit $j \geq i+k+2$ und $i \geq j+k+2$ (mod n). Der Teilgraph $D_{1k}(n)$ von G zerfällt dann in 2 Zusammenhangskomponenten. Zu zeigen ist, daß der Teilgraph D(n) von G diese beiden Komponenten verbindet.
Sei ohne Einschränkung j>i+k (sonst i>j+k). Dann gilt $u_r := u_{i+k+\lfloor (j-i-k)/2 \rfloor} \notin V$ und $u_{r+n/2 \pmod n} \notin V$, so daß $(u_r,u_{r+n/2}) \in E_{D(n)}$ gilt. Diese beiden Knoten sind Vertreter der beiden Zusammenhangskomponenten im Restgraphenanteil $D_{1k}(n)$. Der Restgraph ist damit zusammenhängend.
Fall b) Fall a) tritt nicht ein: Der Restgraph ist dann zusammenhängend, da schon der Teilgraph $D_{1k}(n)$ nach Entfernung der t-1 Knoten zusammenhängend bleibt.
Der Beweis für die anderen Fälle von G erfolgt in analoger Weise. q.e.d.

Ein Diagnosealgorithmus für Modell VT1 wird in [18] vorgestellt. Er baut auf dem Satz auf, daß es in einem t-diagnostizierbaren Diagnosegraphen G=(N,E), falls zwischen einer und t Einheiten defekt sind und damit nicht das 1-Syndrom vorliegt, stets einen Knoten u∈N mit der Eigenschaft

$$|\{v \in N \mid u \text{ intakt} \Rightarrow v \text{ defekt}\}| > t$$

gibt. Wgen der t-Fehlerannahme muß dann u defekt sein. Der Algorithmus von Chwa und Hakimi setzt nun in einem Diagnoseschritt alle derartigen Knoten defekt, entfernt sie aus dem Diagnosegraphen und vermindert den Wert t um die Zahl dieser Knoten. Im verbleibenden Restgraphen, der t-diagnostizierbar für den nun aktuellen Wert von t ist, wird entsprechend weiter verfahren. Chwa und Hakimi geben für ihren Algorithmus eine Komplexität von $O(|E|)$ an, in der Formulierung von [18] werden aber mindestens $O(|N|^2)$ Schritte benötigt.

## 4.3. DAS VERGLEICHSTESTMODELL VT2

Dieses Modell macht die Annahme, daß die Vergleichstestergebnisse zweier defekter Einheiten niemals Übereinstimmung bringen. Also muß ausgeschlossen sein, daß zwei Einheiten, die durch eine Testverbindung einander vergleichen, zur selben Zeit in derselben Weise fehlerhaft sind. Außerdem müssen die verwendeten Testprogramme so umfangreich und ausgeklügelt sein, daß jeder möglicher Fehler einer Einheit zu einer charakteristischen Änderung des Verhaltens der Einheit als Antwort auf dieses Testprogramm führt.

Da in Modell VT2 ein Testergebnis 1 stets anzeigt, daß beide am Test beteiligten Einheiten intakt sind, können bei einem vorliegenden Syndrom Teile des Systems bzw. des Diagnosegraphen auf folgende einfache Art diagnostiziert werden: Ergibt die Kante $(u_i,u_j) \in E$ das Vergleichsergebnis 1, so sind $u_i$ und $u_j$ intakt; existieren weiterhin mit 0 bewertete Kanten $(u_i,u_k) \in E$ bzw. $(u_j,u_k) \in E$, so ist $u_k$ defekt. Dieses einfache Vorgehen wird explizite Diagnose genannt.

Definition: Ein Diagnosegraph G heiße stark t-diagnostizierbar in Modell VT2, falls unter der Voraussetzung, daß die Annahmen des Modells VT2 gelten und nicht mehr als t Einheiten defekt sind, aus der Kenntnis des Syndroms allein durch explizite Diagnose alle defekten Einheiten bestimmt werden können.
G heiße stark t-optimal in Modell VT2, falls G stark t-diagnostizierbar und t-optimal in Modell VT2 ist.

Die Struktur t-diagnostizierbarer Diagnosegraphen wird durch den folgenden Satz geklärt. Formulierung und Beweis erfolgen in Analogie zu Theorem 2 in [11], das die entsprechende Aussage für das BGM-Modell macht.

SATZ 4.2.
Ein Diagnosegraph G=(N,E) ist t-diagnostizierbar in Modell VT2 genau dann, wenn gilt:

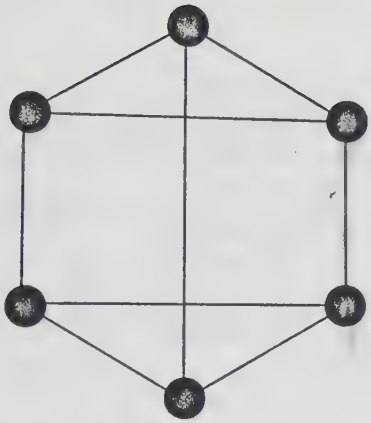
1) $d_{min} \geq t$
2) Für jede Kante $(u,v) \in E$ mit $d(u)=d(v)=t$ existiert ein $w \in N$, so daß entweder $w \in N(u) \setminus N(v)$ und $N(w) \neq N(v)$ gilt
   oder $w \in N(v) \setminus N(u)$ und $N(w) \neq N(u)$.

Beweis:
Im Beweis von Theorem 2 in [11] ersetzt man die Vorgängermenge $B(x)$ von $x \in N$ durch die Nachbarmenge $N(x)$ sowie gerichtete Tests durch ungerichtete. Für $X \subseteq N$ mit $|X| \geq 2$ ist außerdem $\Delta(X)$ als $N(X)$ und $B(X)$ als (unechte) Nachbarmenge $\bigcup_{x \in X} N(x)$ zu verstehen.

q.e.d.

Direkt aus Satz 4.2. folgt die Gültigkeit von
Korollar 4.1.:
Sei G=(N,E) ein Diagnosegraph. Dann gilt
1) G t-diagnostizierbar in VT2 $\Rightarrow t \leq |N|-2$ und $|E| \geq \lceil |N| \cdot t/2 \rceil$.
2) G t-diagnostizierbar in VT2 und $|E| = \lceil |N| \, t/2 \rceil$
   $\Longrightarrow$ G t-optimal in VT2.
3) $d_{min} \geq t+1 \Longrightarrow$ G t-diagnostizierbar in Modell VT2.

Figur 4.4. zeigt ein Beispiel: $G^+$ ist 3-diagnostizierbar in Modell VT2, sogar 3-optimal.



Figur 4.4. $G^+$

Im weiteren Verlauf dieses Abschnitts 4.3. wird unter (starker) t-Diagnostizierbarkeit und (starker) t-Optimalität stets die entsprechende Eigenschaft in Modell VT2 verstanden.

Mit kleineren Änderungen kann der Strukturalgorithmus, der für das BGM-Modell in Abschnitt 3.2. angegeben wurde, auch für das VT2-Modell dienen (man ersetze einfach die Vorgängermengen V(x) durch die Nachbarmengen N(x)).

Ein Diagnosealgorithmus für VT2 wurde in [4] ausführlich vorgestellt. In seiner seriellen Version benötigt er $O(|N|^2)$ Schritte bei Verwendung einer Adjazenzmatrix zur Darstellung des Diagnosegraphen. Ebenfalls vorgestellt wurde eine parallele Version dieses Algorithmus, die auf der sogenannten Baummaschine [12,13] abläuft und $O(|N|)$ Schritte benötigt (Mengenoperationen werden jeweils als Einzelschritte gezählt). Die Wahl der Baummaschine als Modell einer Maschine, auf der der parallele Algorithmus implementiert werden kann, ist dabei motiviert durch die Möglichkeit, sie effektiv in VLSI-Technologie zu verwirklichen [61].

Hier nun eine knappe Beschreibung der seriellen Version des Diagnosealgorithmus:

Schritt 1: Explizite Diagnose.
Updaten (siehe unten).
Falls t=0 ist, werden alle noch nicht diagnostizierten Knoten intakt gesetzt.

Schritt 2: Falls $N \neq \emptyset$ und $t > 0$ ist:
Alle $u \in N$ mit $d(u) \neq t$ sind defekt.
Updaten (siehe unten).

Schritt 3: Falls $N \neq \emptyset$ und $t > 0$ ist:
Berechne $N(u)$ für alle $u \in N$.
Berechne die eindeutige Menge $N' \subseteq N$ mit $|N'| = |N| -$
in der alle Elemente $u \in N'$ dieselbe Nachbarmenge $N(u)$ haben.
Alle $u \in N'$ sind intakt, alle $u \in N \setminus N'$ defekt.

Dabei werden unter "Updaten" die folgenden Anweisungen verstanden:

$N := N \setminus \{$diagnostizierte Knoten in diesem Schritt$\}$.
$n := |N|$ .
$t := t - |\{$in diesem Schritt gefundene defekte Knoten$\}|$
Alle Einträge, die zu diagnostizierten Knoten gehören, werden aus der Adjazenzmatrix gelöscht.

Die Richtigkeit dieses Diagnosealgorithmus wird durch die folgenden Überlegungen sichergestellt. Wird in Schritt 2 ein noch nicht diagnostizierter Knoten u mit mehr als t ebenfalls noch nicht diagnostizierten Knoten verglichen ($d(u) < t$ kann nicht eintreten), so ist u defekt, da alle diese Vergleiche das Ergebnis 0 erbracht haben und höchstens t der noch nicht diagnostizierten Knoten defekt sein können.
In Schritt 3 ergibt das Auffassen von $N'$ als Menge der intakten und $N \setminus N'$ als Menge der defekten Knoten Konsistenz mit dem vorliegenden Syndrom. Wegen der t-Diagnostizierbarkeit von G existiert aber nur genau ein konsistentes Fehlermuster mit höchstens t Elementen.

Stark t-diagnostizierbare Diagnosegraphen benötigen nur Schritt 1 dieses Algorithmus. Beispiele solcher Graphen sind Diagnosegraphen G mit $d_{min} \geq t+1$. Im folgenden soll nun die Klasse der t-optimalen Diagnosegraphen näher untersucht werden. Daß für alle möglichen Werte $n = |N| \geq 3$ und $1 \leq t \leq n-2$ t-optimale Graphen existieren, zeigt dabei Figur 4.5. bzw. Proposition 4.4..

| Parameter | Diagnosegraph | Kantenzahl |
|---|---|---|
| t=1, n≥4 | TK(n) | n-1 |
| t=2, n≥5 | $D_{11}(n)$ | n |
| t=3, n=6 | $G^+$ | 9 (=nt/2) |
| 2<t<n-2, n≥7 | | |
| a) t=2k | $D_{1k}(n)$ | kn = nt/2 |
| b) t=2k+1 | | |
| n gerade | $D_{1k}(n) \cup D(n)$ | nt/2 |
| n ungerade | $D_{1k}(n) \cup S(n)$ | $\lceil nt/2 \rceil$ |
| t=n-2 | $K_n^-$ | n(n-1)/2 -1 |

( $D_{1k}(n)$ und D(n) bzw. $D_{1k}(n)$ und S(n) haben identische Knotenmengen )

Figur 4.5. t-optimale Diagnosegraphen

Proposition 4.4.
Alle in Figur 4.5. aufgeführten Diagnosegraphen sind t-optimal, mit Ausnahme von $K_n^-$ sogar stark t-optimal.
Unter $K_n^-$ wird dabei der vollständige Graph mit n Knoten [36] verstanden, aus dem eine beliebige Kante entfernt wurde.

Beweis: Hier sei exemplarisch die Gültigkeit dieser Aussage für $K_n^-$ und für die Graphen $D_{1k}(n) \cup D(n)$ gezeigt.
1) $K_n^-$ ist (n-2)-diagnostizierbar (n≥3):
Mit Ausnahme zweier Knoten u und v gilt d(w)=n-1≥t+1 für alle w∈N. Für diese Knoten u und v gilt zwar d(u)=d(v)=t, aber (u,v)∉E. Damit sind die Bedingungen aus Satz 4.2. erfüllt.
2) $K_n^-$ ist (n-2)-optimal:
Angenommen, ein Graph G=(N,E) mit n Knoten und $|E| < \frac{n(n-1)}{2} - 1$ sei t-diagnostizierbar. Dann gilt d(u)≥t=n-2 für alle u∈N und

es existieren mindestens zwei Paare $(u,v)$, $(u',v') \notin E$. Wegen $d_{min} \geq n-2$ sind diese Paare knotendisjunkt. Dann aber erzeugen die beiden Fehlermuster $N \setminus \{u,v\}$ und $N \setminus \{u',v'\}$ das O-Syndrom und G ist nicht $(n-2)$-diagnostizierbar.

3) $K_n^-$ ist nicht stark t-optimal:

Sind u und v die beiden Knoten mit Grad n-2, so erzeugt das Fehlermuster $F=N \setminus \{u,v\}$ das O-Syndrom, so daß durch explizite Diagnose kein Knoten diagnostiziert werden kann.

4) $D_{1k}(n) \cup D(n)$ $(n \geq 8, 1 \leq k \leq (n-4)/2)$ ist t-diagnostizierbar $(t=2k+1)$:

Es gilt $d_{min}=t$. In diesem Beweisteil stehen nun Indizes i,j.. für die Einheiten $u_i, u_j$.. und die Rechnungen mit Indizes sind modulo n zu verstehen.

Seien also $i,j \in N$ und $(i,j) \in E$ (dann $d(i)=d(j)=t$). Dann gilt $N(i) = \{i+1, i+2, .., i+k, i-1, ..., i-k\} \cup \{i+n/2\}$, entsprechend N(j). Weiter $|N(j) \cup \{j\}| = t+1 \leq n-2$. Setze $m := n-(t+1)$, m ist gerade. Dann gilt $j+n/2 \pm r \notin N(j)$ für alle $1 \leq r \leq m/2$. Es ist aber $A := \{j+n/2+r \mid 1 \leq r \leq m/2\} \cap N(i) \neq \emptyset$ und für jedes $s \in A$ gilt $N(s) \cap A \neq \emptyset$ (für $m/2 \geq 2$ gilt $s+1 \in A$ oder $s-1 \in A$, für $m/2=1$ ist $t=n-3$ und $k \geq 2$, also $s+2 \in A$ oder $s-2 \in A$).

Damit existiert also ein $s \in N(i) \setminus N(j)$ mit $N(s) \cap A \neq \emptyset$, wegen $N(j) \cap A = \emptyset$ also $N(s) \neq N(j)$ und Bedingung 2) aus Satz 4.2. ist erfüllt; $D_{1k}(n) \cup D(n)$ ist $(2k+1)$-diagnostizerbar.

5) $D_{1k}(n) \cup D(n)$ ist t-optimal $(t=2k+1)$:

Die Kantenzahl ist nt/2 wegen $d(u)=t$ für alle $u \in N$. Nach Korollar 4.1.2) ist damit die Optimalität zusammen mit Beweisteil 4) gezeigt.

6) $G=D_{1k}(n) \cup D(n)$ ist stark t-optimal:

Bei Auftreten eines Fehlermusters $F=\{u_{i_1}, \ldots, u_{i_t}\}$ von t Elementen kann in G das O-Syndrom nicht auftreten. Wegen der Symmetrie von G würde sonst nämlich auch das von F verschiedene Fehlermuster $F'=\{u_{i_1+1}, \ldots, u_{i_t+1}\}$ das O-Syndrom hervorrufen im Widerspruch zur t-Diagnostizierbarkeit von G.

Fall a) Alle intakten Einheiten in G sind bei Auftreten von höchstens t defekten Einheiten durch explizite Diagnose feststellbar. Wegen $d_{min}=t$ wird dann auch jede defekte Einheit mit mindestens einer dieser intakten Einheiten verglichen und durch die explizite Diagnose als defekt erkannt.

Fall b) Für eine intakte Einheit $u_i \in N$ bestehe $N(u_i)$ gerade aus

den defekten Einheiten. Seien hier wieder Einheiten durch ihre Indizes bezeichnet. Dann ist $N(i)=\{i+1,..,i+k,i-1,..,i-k\}\cup\{i+n/2\}$ das vorliegende Fehlermuster. Alle intakten Einheiten (außer i) sind durch explizite Diagnose als solche feststellbar, denn: Ist $m:=n-t+1>2$, so sind die Kanten $(i+n/2+1,i+n/2+2),...$ $(i-k-2,i-k-1),(i+n/2-1,i+n/2-2),...,(i+k+2,i+k+1)$ alle mit 1 bewertet; ist dagegen $m=2$, so $k\geq 2$ wegen $n\geq 7$ und es gilt $(i+n/2+1,i+n/2-1)\in E$ und diese Kante ist mit 1 bewertet. In beiden Fällen werden alle Elemente aus N(i) durch die explizite Diagnose erfaßt. q.e.d.

Die Klasse der t-optimalen Graphen in Modell VT2 wird nun genauer untersucht. Dabei bezeichne

Opt(n,t) die Menge der t-optimalen Graphen mit n Knoten,

$O(n,t)\subseteq Opt(n,t)$ die Menge der t-optimalen Graphen, die bei einem Fehlermuster mit höchstens t Elementen das O-Syndrom zulassen ,

$N(n,t) := Opt(n,t)\setminus O(n,t)$,

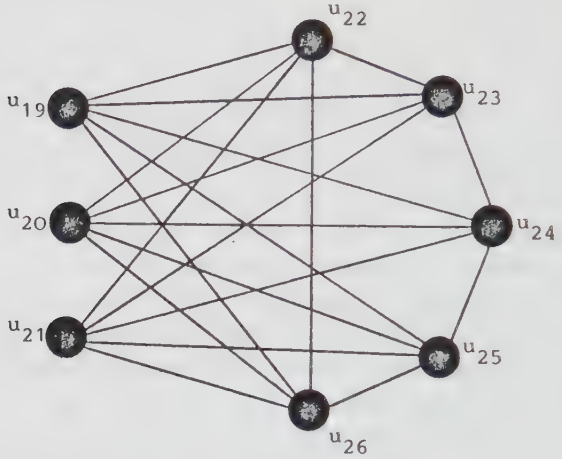
SN(n,t) die Menge der stark t-optimalen Graphen mit n Knoten.

Diese Einteilung erfolgt im Hinblick auf die Einfachheit der Diagnose für die jeweiligen Diagnosegraphen: Elemente aus SN(n,t) werden stets durch die explizite Diagnose vollständig diagnostiziert, Elemente aus N(n,t) lassen das O-Syndrom nicht zu und werden deshalb durch die explizite Diagnose wenigstens teilweise diagnostiziert, während für Elemente in O(n,t) die explizite Diagnose völlig nutzlos sein kann.

Durch Proposition 4.4. wurde bereits gezeigt, daß $SN(n,t)\neq\emptyset$ ist für alle zulässigen Parameter n und t mit $t<n-2$. $SN(n,n-2)=\emptyset$ gilt, weil für $t=n-2$ die minimale Kantenzahl $n(n-1)/2-1$ beträgt (da $K_n^-$ $(n-2)$-optimal ist) und da bei Auftreten von n-2 Fehlern dann das O-Syndrom möglich wird.
Die naheliegende Vermutung $N(n,t)=SN(n,t)$ für alle möglichen n und t wird durch das folgende Gegenbeispiel widerlegt. Ein Diagnosegraph $G_{26}$ mit 26 Knoten wird konstruiert aus den Bestandteilen

$D_{12}(8)$ mit Knotenmenge $\{u_1,u_2,..,u_8\}$, $A_5$ mit Knotenmenge $\{u_9,...,u_{13}\}$, $A_1$ mit Knotenmenge $\{u_{14}\}$, $A_4$ mit Knotenmenge $\{u_{15},...,u_{18}\}$ und $G_1$ mit Knotenmenge $\{u_{19},...,u_{26}\}$.

Dabei sind die Graphen $A_i$ als triviale Graphen mit i Knoten und keiner Kante definiert, der Graph $G_1$ wird in Figur 4.6. gezeigt.




Figur 4.6. Der Diagnosegraph $G_1$

Formal besteht $G_1$ aus einem Verbund $A_3+D_{11}(5)$, wobei $A_3=(\{u_{19},u_{20},u_{21}\},\emptyset)$ ist. Der Graph $G_{26}$ ist nun definiert als

$G_{26}:=(D_{12}(8)+A_5)\cup(A_5+A_1)\cup(A_1+A_4)\cup(A_4+G_1)$.

Mit Hilfe von Satz 4.2. ergibt sich die 9-Diagnostizierbarkeit von $G_{26}$, wegen $d(u)=9$ für alle Knoten u auch die 9-Optimaltität. Es gilt $G_{26}\in N(26,9)\setminus SN(26,9)$, da bei Auftreten des Fehlermusters $F=\{u_{15},u_{16},u_{17},u_{18},u_{22},u_{23},u_{24},u_{25},u_{26}\}$ die dem Graphen $G_1$ entsprechenden Knoten $u_{19},\ldots,u_{26}$ durch explizite Diagnose nicht erfaßt werden. Daß $G_{26}\in N(26,9)$ gilt, ist leicht einzusehen, da zur Entstehung des O-Syndroms im Teilgraph $D_{12}(8)$ bereits mindestens 6 Knoten defekt sein müssen und im Teilgraph $G_1$ mindestens 5. Unser nächstes Ziel ist die Untersuchung von $O(n,t)$.

Proposition 4.5.:
Sei $G=(N,E)\in O(n,t)$, $n\geq 6$ und $t\geq 2$.
Dann gilt $(n+1)/2<t\leq n-2$.

Beweis:
Sei F das eindeutige Fehlermuster mit höchstens t Elementen, das das O-Syndrom erzeugt. Wegen $t\geq 2$ gilt nach Proposition 4.4. bzw. Figur 4.5. $|E|=\lceil nt/2\rceil$.
Fall a) $nt/2\in\mathbb{N}$: Dann gilt
i) $\forall\, u\in N\setminus F\ \forall\, v\in F$: $(u,v)\in E$ wegen $d(u)=t$
ii) $\forall\, u,v\in N\setminus F$: $(u,v)\notin E$.
Durch i) sind $(n-t)t$ Kanten festgelegt. Aus $(n-t)t\leq |E|=nt/2$ folgt sofort $t\geq n/2$. Da sonst F und $N\setminus F$ das O-Syndrom erzeugten, ist $t=n/2$ nicht möglich, also $t>n/2$. Wegen $n\geq 6$ ist also $t\geq 4$. Falls n gerade ist, folgt aus $t>n/2$ auch $t>(n+1)/2$. Ist n ungerade, würde aus $n/2<t=(n+1)/2$ die Beziehung
$|E|-(n-t)t=nt/2-(n-t)t=t/2$ und $n-t=t-1$ folgen. Für alle $u\in F$ gäbe es dann genau ein $v\in F$ mit $(u,v)\in E$. Dann erzeugten aber die beiden verschiedenen Fehlermuster $F_1=(N\setminus F)\cup\{u\}$ und $F_2=(N\setminus F)\cup\{v\}$ $(u,v\in F,(u,v)\in E)$ dasselbe Syndrom. Also gilt $t>(n+1)/2$.
Fall b) $nt/2\notin\mathbb{N}$, n und t sind ungerade Zahlen. Dann existiert genau ein $x\in N$ mit $d(x)=t+1$, x ist defekt.
Analog zu Fall a) wird nun auf $(n-t)t\leq|E|=\lceil nt/2\rceil=(nt+1)/2$ geschlossen, woraus $t(2t-n)\geq -1$ $(*)$ folgt. Wegen $n\geq 6$ ist $t=1$ nicht möglich. Ein Wert $t\geq 2$ hat aber wegen $(*)$ $2t-n\geq 0$ und deshalb $t\geq n/2$ zur Folge. Also gilt $t\geq(n+1)/2$, da n ungerade ist.
Angenommen $t=(n+1)/2$. Da t ungerade sein muß, folgt daraus $n\equiv 1 \pmod 4$, etwa $n=4k+1$ $(k\in\mathbb{N})$ und $t=2k+1$. Dann gilt $|E|-(n-t)t=k+1$. Wegen $n\geq 6$ gilt $k\geq 2$ und deshalb $t\geq 5$. Wegen des O-Syndroms gilt sicher $|F|=t$. Also existieren $k+1\geq 3$ Kanten im von F aufgespannten Teilgraph von G, es existiert ein Paar $u,v\in F$ mit $(u,v)\in E$ und $(u,w)\notin E$, $(v,w)\notin E$ für alle $w\in F\setminus\{u,v\}$.
Das analoge Argument wie in Fall a) schließt nun den Beweis ab.

q.e.d.

Proposition 4.6.:
Sei $(n+1)/2<t<n-2$ (d.h. $n\geq 8$) und sei $G=(N,E)$ ein Diagnosegraph. Dann gilt $G\in O(n,t)$ genau dann, wenn $G=A_{n-t}+G_t$ gilt, wobei $G_t\in N(t,2t-n)$ ist und $G_t$ und $A_{n-t}$ disjunkte Knotenmengen haben.

Beweis:
1) Die Notwendigkeit: Sei $G\in O(n,t)$ und $(n+1)/2<t<n-2$, sei F das eindeutige Fehlermuster mit t Elementen, das das O-Syndrom erzeugt und sei $G_t$ der Teilgraph von G, der durch F aufgespannt wird.
Wegen des O-Syndroms ist dann der von $N\setminus F$ aufgespannte Teilgraph von G der triviale Graph $A_{n-t}$. Gilt $nt/2\notin\mathbb{N}$, gehört der eindeutige Knoten mit Grad t+1 zu F, alle $u\in N\setminus F$ haben Grad t.
Deshalb folgt $G=A_{n-t}+G_t$, zu zeigen bleibt $G_t\in N(t,2t-n)$.
Sei $G_t=(N_t,E_t)$, also $N_t=F$. Für alle $u\in N_t$ gilt
$|N_t(u)| = t-(n-t)=2t-n$ ( $N_t(u)$ Nachbarmenge von u in $G_t$)
(mit einer Ausnahme u für $nt/2\notin\mathbb{N}$, dieser Knoten erfüllt $|N_t(u)| = 2t-n+1$). Stets gilt also $|E_t| = \lceil t(2t-n)/2\rceil$ (man beachte, daß für $nt/2\notin\mathbb{N}$ $t(2t-n)/2\notin\mathbb{N}$ gilt).
Angenommen, $G_t$ ist nicht (2t-n)-diagnostizierbar. Dann existieren verschiedene Fehlermuster $F_1$ und $F_2$ mit höchstens 2t-n Elementen, die dasselbe Syndrom erzeugen. Dann erzeugen aber auch die verschiedenen Fehlermuster $(N\setminus F)\cup F_1$ und $(N\setminus F)\cup F_2$ dasselbe Syndrom in G, in Widerspruch zur t-Diagnostizierbarkeit von G.
Schließlich ist das O-Syndrom in $G_t$ bei höchstens 2t-n Fehlern nicht möglich. Wäre nämlich F' ein solches Fehlermuster, erzeugte $(N\setminus F)\cup F'$ in G ebenso wie F das O-Syndrom.
2) Die angegebene Bedingung ist auch hinreichend:
Sei $G=A_{n-t}+G_t$ und $G_t=(N_t,E_t)\in N(t,2t-n)$. Wegen $(n+1)/2<t<n-2$ gilt $2t-n\geq 2$ und deshalb $|E_t| = \lceil t(2t-n)/2\rceil$ (vgl. Proposition 4.4.).
Zu zeigen ist $G\in O(n,t)$.
i) Das O-Syndrom wird durch das Fehlermuster $F=N_t$ erzeugt.
ii) Für alle $u\in N$ gilt $d(u)=t$ (mit einer Ausnahme für $nt/2\notin\mathbb{N}$, dieser Knoten hat Grad t+1 und gehört zu $N_t$).
iii) Es gilt Bedingung 2) aus Satz 4.2.:
Seien $u,v\in N$ mit $d(u)=d(v)=t$ und $(u,v)\in E$.
Fall a) $u\in N\setminus N_t$ und $v\in N_t$: Dann gilt $N(u)=N_t$ und $N(v)=(N\setminus N_t)\cup N_t(v)$. Also existieren $w_1,w_2\in N_t$ mit $(w_1,w_2)\in E$ und $w_1,w_2\notin N(v)\cup\{v\}$, da sonst $F'=N_t(v)$ in $G_t$ das O-Syndrom erzeugte; aber $G_t\in N(t,2t-n)$.

Fall b) $u,v \in N_t$: Wegen $|N_t(u)| = |N_t(v)| = 2t-n$ und da $G_t$ $(2t-n)$-diagnostizierbar ist, existiert ohne Einschränkung ein $w \in N_t(u) \setminus N_t(v)$ mit $N_t(w) \neq N_t(v)$ (Satz 4.2.). Dann gilt aber auch $w \in N(u) \setminus N(v)$ und $N(w) \neq N(v)$ und G ist t-diagnostizierbar. q.e.d.

Damit ist gezeigt, daß für $(n+1)/2 < t < n-2$ $O(n,t) \neq \emptyset$ gilt, da $N(t,2t-n)$ nach Proposition 4.4. nicht leer ist. Wegen $K_n^-$ ist auch $O(n,n-2) \neq \emptyset$ für alle $n \geq 3$.
Nach Proposition 4.5. gilt $O(n,t) = \emptyset$ für $2 \leq t \leq (n+1)/2$ und $n \geq 6$. Leicht überlegt man sich $O(5,2) = \emptyset$; die Aussage $O(n,1) \neq \emptyset$ wird durch die folgenden Beispiele belegt:

Für $n \geq 4$ werde der Graph $G_1 = (N_1, E_1)$ mit $|N_1| = n$ durch $E_1 := \{(u_1, u_i) \mid i = 2,3,..,n\}$ definiert ($G_1$ ist der sogenannte Sterngraph mit n Knoten). $G_1$ ist 1-optimal und das Fehlermuster $F = \{u_1\}$ erzeugt das O-Syndrom, also $G_1 \in O(n,1)$.

Es gilt $Opt(n,t) = O(n,t) \dot{\cup} (N(n,t) \setminus SN(n,t)) \dot{\cup} SN(n,t)$.
Graphentheoretische Eigenschaften wie t-Diagnostizierbarkeit, t-Optimalität, starke t-Diagnostizierbarkeit, starke t-Optimalität, Element aus $O(n,t)$, $N(n,t)$ oder $SN(n,t)$ werden durch Graphenisomorphismen [36] respektiert. Also sind die angegebenen Mengen jeweils Vereinigungen von Isomorphieklassen.

Anmerkung: Ein Vergleichstestmodell, das VT2 ähnelt, wurde mit etwas anderen Annahmen unabhängig in [53] eingeführt.

Zum Abschluß dieses Abschnitts soll genauer auf die Komplexitätsfrage der Diagnose im VT2-Modell eingegangen werden. Der schon beschriebene Diagnosealgorithmus für dieses Modell baut auf einem t-diagnostizierbaren Graphen und der Annahme von höchstens t Fehlern auf und arbeitet in polynomialer Zeit $O(|N|^2)$. Ein Abschwächung dieser Annahmen führt zum folgenden Problem:

> Komplexität der Diagnose in VT2, falls ein beliebiger Diagnosegraph benutzt wird und keine Annahme über die Zahl der Fehler gemacht wird.

Diagnose bedeutet hier die Bestimmung eines minimalen konsistenten Fehlermusters (die Eindeutigkeit ist nicht mehr gewährleistet). Dieses ist ein sogenanntes NP-schwieriges Problem [34], da bei Auftreten etwa des O-Syndroms ein minimales konsistentes Fehlermuster gerade eine minimale Knotenüberdeckung (vertex-cover)

ist und das "Vertex-Cover"-Problem NP-vollständig ist [34].

## 4.4. DAS VERGLEICHSTESTMODELL VT3

Modell VT3 macht die zunächst verblüffende Annahme, daß zwei defekte Einheiten immer auf dieselbe Weise auf ein Testprogramm reagieren. Folgende Anwendung dieses Modells läßt sich in einer Zeit der Massenproduktion von Mikroprozessoren vorstellen. Eine Produktionsserie von VLSI-Chips soll auf einen bestimmten systematischen Fehler, der sich während der Produktion zeitweise ergeben haben könnte, untersucht werden. Dann ist klar, daß sich alle intakten Chips identisch verhalten und daß alle defekten Chips auf ein auf den systematischen Fehler zugeschnittenes Testprogramm in derselben Weise reagieren.

SATZ 4.3.
Ein Diagnosegraph $G=(N,E)$ ist t-diagnostizierbar in Modell VT3 genau dann, wenn $t \leq (|N|-1)/2$ gilt.

Beweis:
Wäre $t \geq |N|/2$ und $F \subset N$ mit $|F|=t$, so erzeugten F und $N \setminus F$ nach den Annahmen von VT3 dasselbe Syndrom. Wegen $|F|=t$ und $|N \setminus F| \leq t$ wäre dann G nicht t-diagnostizierbar in VT3. Daß $t \leq (|N|-1)/2$ auch hinreichend für die t-Diagnostizierbarkeit von G in VT3 ist, läßt sich dem folgenden Diagnosealgorithmus entnehmen. q.e.d.

Ein Diagnosealgorithmus für Modell VT3

Schritt 1: Suche in G einen wegen des Zusammenhangs von G existierenden aufspannenden Baum mit Hilfe der Tiefensuche [71].
Numeriere die Knoten entsprechend ihres Auftretens durch.

Schritt 2: Setze den Knoten mit Nummer 1 intakt.
Sind schon die ersten i Knoten diagnostiziert ($1 \leq i \leq n-1$), so setze den (i+1)-ten Knoten auf denselben Zustand wie sein bereits diagnostizierter Nachbarknoten, falls der Vergleich das Ergebnis 1 gebracht hat; sonst auf den entgegengesetzten Zustand.

Schritt 3: Falls in Schritt 2 mehr als t defekte Knoten notiert worden sind, werden sämtliche Zustandszuweisungen umgekehrt.

Wegen $t \leq (|N|-1)/2$ liefert dieser Algorithmus genau ein zum vorliegenden Syndrom konsistentes Fehlermuster mit höchstens t Elementen. Schritt 1 benötigt $O(|E|)$ Schritte, die Schritte 2 und 3 jeweils $O(|N|)$. Wird von vornherein nur ein Diagnosebaum oder sogar eine Diagnosekette als Diagnosegraph vorgesehen, entfällt Schritt 1 und die Diagnose gelingt in linearer Zeit.

Ein Strukturalgorithmus für Modell VT3 ist nicht notwendig. Für einen Diagnosegraphen mit n Knoten stellt stets $t := \lfloor (n-1)/2 \rfloor$ den höchsten möglichen Grad der Diagnostizierbarkeit dar. Optimale Diagnosegraphen sind Diagnosebäume.

## 4.5. ANMERKUNGEN ZU DEN VERGLEICHSTESTMODELLEN

a) Bei einem Vergleich der Struktursätze der gerichteten Modelle PMC, BGM und A1 auf der einen Seite und der ungerichteten Modelle VT1, VT2 und VT3 auf der anderen Seite fällt eine weitgehende Analogie auf (beim Paar A1 - VT3 mache man sich klar, daß die analoge Bedingung 2) in Satz 3.3. für das VT3-Modell für zusammenhängende Graphen stets erfüllt ist): Die Terme Vorgängermenge und Nachfolgermenge sind einfach durch Nachbarmenge zu ersetzen, der Eingangsgrad durch den Grad eines Knotens.

Definition: Ist G ein gerichteter Diagnosegraph ohne Einzeltests, so entstehe der ungerichtete Graph $G_u$ aus G durch Ersetzen jedes Doppeltests durch einen ungerichteten Test.

Proposition 4.7.:
Sei G ein gerichteter Diagnosegraph ohne Einzeltests. Dann gilt:

$$G \text{ t-diagnostizierbar in Modell } \left\{\begin{matrix}\text{PMC}\\ \text{BGM}\\ \text{A1}\end{matrix}\right\} \iff$$

$$G_u \text{ t-diagnostizierbar in Modell } \left\{\begin{matrix}\text{VT1}\\ \text{VT2}\\ \text{VT3}\end{matrix}\right\} .$$

Beweis:
Verwende die Struktursätze 3.1.,3.2.,3.3.,4.1.,4.2. und 4.3.. Dabei ist der minimale Eingangsgrad von G dem minimalen Grad von $G_u$ gleich und die Vorgänger- bzw. Nachfolgermengen in G stimmen

mit den Nachbarmengen in $G_u$ überein, da G nur Doppeltests hat. Die Bedingung 2) in Satz 3.3. ist für zusammenhängende Graphen ohne Einzeltests stets erfüllt q.e.d.

b) Bisher wurde noch nichts über das Problem gesagt, wie die Diagnosegraphen in Systemen aufgebaut werden. Bei Verwendung von Vergleichstestmodellen etwa kann eine Einheit zu einem Zeitpunkt nur mit genau einer anderen Einheit verglichen werden. In den Modellen VT1 und VT2 muß aber in einem t-diagnostizierbaren Graphen jede Einheit mit mindestens t anderen Einheiten verglichen werden.
In Kapitel 2 wurde die k-Konstruierbarkeit ( = k-Färbbarkeit für Graphen) definiert. Bezeichnet KF(G) die Kantenfärbbarkeit des Diagnosegraphen G, so werden mindestens KF(G) Testrunden benötigt, um ein vollständiges Syndrom zu errechnen.

SATZ 4.4. (Theorem 14.4.1. und 14.4.2. aus [64])
Ist G=(N,E) ein (ungerichteter) Graph und bezeichnet $d_G := \max_{u \in N} d(u)$ den maximalen Grad von G, so gilt $d_G \leq KF(G) \leq d_G+1$.

Für die Modelle VT1 und VT2 wurden t-optimale Diagnosegraphen für alle jeweils möglichen Parameter n angegeben (Figur 4.3. und Figur 4.5.). Für diese Graphen gilt

$$d_G = \begin{cases} t & \text{falls } t \geq 2 \text{ und } nt/2 \in \mathbb{N}. \\ t+1 & \text{falls } t=1 \text{ oder } nt/2 \notin \mathbb{N}. \end{cases}$$

Die höchstens notwendige Zahl von Vergleichsrunden zur Berechnung eines Syndroms beträgt t+1 bzw. t+2, ist also linear in t.

Ein Vorschlag zum Aufbau von Vergleichsgraphen nach [23]: Die durch den Vergleichsgraphen festgelegten Vergleiche werden ausgeführt, indem jeder Job von einem Paar von (im Graphen adjazenten) Prozessoren abgearbeitet wird und die Ergebnisse verglichen werden. Zu einem Zeitpunkt können so höchstens $\lfloor |N| /2 \rfloor$ Jobs berechnet werden. Die Diagnose kann durchgeführt werden, sobald alle Vergleiche durch derartige Jobabarbeitungen realisiert worden sind.

c) Die Suche nach einem Diagnosegraphen, der t-diagnostizierbar etwa in Modell VT2 sein soll, läßt sich auch als das folgende "nicht-sequentielle Suchproblem" [2] beschreiben:

Gegeben: Die Parameter n und t.
Gesucht: Eine Menge $E \subseteq \{(i,j) \mid i,j \in \{1,\ldots,n\}\}$ von Paaren mit der folgenden Eigenschaft:
Sind n (Ergebnis-) Werte $a(1),\ldots,a(n)$ vorgegeben, von denen mindestens n-t übereinstimmen und die restlichen höchstens t Werte von diesem Wert abweichen und paarweise verschieden sind, so läßt sich anhand der durch die Indexpaare von E angegebenen Vergleiche stets die Menge der abweichenden Werte ermitteln.

Man beachte, daß die Menge E der Indexpaare unabhängig von einem konkret vorliegenden Werte-n-tupel $a(1),\ldots,a(n)$ gewählt werden muß.
Eine optimale Lösung dieses Suchproblems ist ein Lösung, für die $|E|$ minimal unter allen Lösungen ist. Mit der beschriebenen Übertragbarkeit der Begriffe "t-Diagnostizierbarkeit" und "nicht-sequentielles Suchproblem mit Parameter t" gilt dann

Proposition 4.8.:
Seien $n \geq 3$, $1 \leq t \leq n-2$ und $G=(N,E)$ ein Diagnosegraph mit $|N| = n$.
Dann gilt:
G ist t-diagnostizierbar in Modell VT2 genau dann, wenn E das oben beschriebene nicht-sequentielle Suchproblem für n und t löst. G ist t-optimal in Modell VT2 genau dann, wenn E eine optimale Lösung dieses Suchproblems ist.

Eine entsprechende Analogie mit einem nicht-sequentiellen Suchproblem läßt sich natürlich auch für die beiden anderen Vergleichstestmodelle beschreiben. Ebenfalls lassen sich Analogien zwischen gerichteten Modellen (etwa den klassischen) und Suchproblemen herstellen.

Thema des folgenden Kapitels ist die dynamische Diagnose, in der die Diagnosestruktur nicht schon von vornherein festgelegt ist. Diese Art der Diagnose entspricht dann der Lösung eines "sequentiellen Suchproblems" [2].

# 5. DYNAMISCHE DIAGNOSE

In den vorangegangenen Kapiteln 3 und 4 wurden ausschließlich Modelle für die sogenannte statische Diagnose vorgestellt, d.h. für eine Diagnose, in der die benutzte Diagnosestruktur (der Diagnosegraph) bereits zu Beginn vollständig festgelegt wird in Abhängigkeit der Zahl der beteiligten Einheiten und der maximalen Zahl der darunter befindlichen fehlerhaften Einheiten. Da dabei zu Beginn natürlich nicht bekannt sein kann, welche Einheiten fehlerhaft sind, muß der Testaufwand einigermaßen gleichmäßig auf das System verteilt werden.

Um den Testaufwand, d.h. die Zahl der benötigten Tests, zu verringern, wird in diesem Kapitel die Methode der dynamischen Diagnose beschrieben. Hier gibt es keine von Anfang an festgelegte Diagnosestruktur, sondern der nächste durchzuführende Test wird in Abhängigkeit der Ergebnisse der bereits durchgeführten Tests festgelegt. Man kann deshalb auch von einer Diagnose mit Gedächtnis sprechen. In der Sprache der Theorie der Suchprobleme entspräche ein solcher Aufbau einer Diagnosestruktur einem sogenannten "sequentiellen Suchproblem" [2].

Bei der statischen Diagnose sind Test- und Diagnosephase deutlich unterscheidbar. Zuerst wird gemäß dem festgelegten Diagnosegraphen getestet und danach die erhaltenen Ergebnisse ausgewertet. Bei der dynamischen Diagnose verschmelzen diese beiden Phasen. Nach jedem durchgeführten Test werden die bereits vorliegenden Ergebnisse ausgewertet und in Abhängigkeit davon wird der nächste durchzuführende Test festgelegt, bis genügend Information für die Gesamtdiagnose vorhanden ist.

Den folgenden Ausführungen sind stets die Modellannahmen des Vergleichstestmodells VT2 zugrunde gelegt. Abschnitt 5.1. stellt drei dynamische Diagnosealgorithmen für dieses Modell vor, berechnet die Zahl der benötigten Tests (Vergleiche) im jeweils ungünstigsten Fall (worst-case-Betrachtung) sowie bestimmt für einen dieser Algorithmen die Zahl der Testrunden (Parallelisierbarkeit) und den Erwartungswert der Zahl der Vergleiche; dies

geschieht jeweils in Abhängigkeit von der Zahl der Einheiten und der maximalen Zahl der fehlerhaften Einheiten.
Abschnitt 5.2. vergleicht die statische und dynamische Diagnose miteinander. Als einziger Vorteil der statischen Diagnose erweist sich dabei die Möglichkeit der dezentralen und lokalen Ausführung, während die dynamische Diagnose deutliche Vorteile hinsichtlich der Zahl der durchzuführenden Vergleiche und der Zahl der durchzuführenden Vergleichsrunden aufweist.

## 5.1. DYNAMISCHE DIAGNOSEALGORITHMEN

Ein Entscheidung für die dynamische Diagnose (Parameter 2.2.2. aus Kapitel 2) legt bereits einen anderen Parameter fest: Die Organisation der Auswertung der Testergebnisse muß global sein, da die Festlegung des nächsten durchzuführenden Tests alle bis dahin gewonnene Information verwendet. Mit den Annahmen des Modells VT2 stellt sich das Problem der dynamischen Diagnose dieser globalen Organisationsinstanz folgendermaßen:

Gegeben: n Werte a(1),...,a(n). Korrekte Werte stimmen überein. Maximal t ($t \leq n-2$) dieser Werte sind inkorrekt. Inkorrekte Werte stimmen paarweise niemals überein.

Gesucht: Die eindeutige Partition $\{1,2,...,n\} = G \dot{\cup} F$, für die gilt:
G = Menge der Indizes der korrekten Werte
F = Menge der Indizes der inkorrekten Werte.

Das bedeutet:
i) $|F| \leq t$ .
ii) $\forall i,j \in G$: $a(i) = a(j)$.
iii) $\forall i,j \in F$: $a(i) \neq a(j)$ $(i \neq j)$ .
iv) $\forall i \in G, j \in F$: $a(i) \neq a(j)$ .

Benutzte Methode: Vergleich von je zwei Werten miteinander. Die Ergebnisse bereits durchgeführter Vergleiche werden bei der Festlegung des nächsten durchzuführenden Vergleichs berücksichtigt.

Im hier verwendeten Modell VT2 darf die Fehlerschranke t den Wert n-2 nicht übersteigen (Korollar 4.1.). Im folgenden werden nun drei Algorithmen für die dynamische Diagnose in diesem Modell vorgestellt. Die Algorithmen A und B können dabei nur für $t \leq (n-1)/2$ eingesetzt werden. Dabei kommt der rekursiv aufgebaute

Algorithmus A meist mit weniger Vergleichen als B aus, während B sehr einfach aufgebaut ist. Algorithmus C schließlich deckt den Bereich $n/2 \leq t \leq n-2$ ab.

Algorithmus A

```
Schritt 0: (Initialisierungen)
           n Anzahl von Werten
           a(1),a(2),..,a(n) Menge der Werte
           t maximale Anzahl von inkorrekten Werten
           (unter Werten werden dabei nur noch nicht diagnos-
            tizierte Werte verstanden)
           Integervariablen i:=1,k:=0,r:=n.
Schritt 1: (Paarbildung)
           WHILE (i≤⌊n/2⌋) AND (k<t) DO
             BEGIN
               Vergleiche a(2i) und a(2i-1) miteinander.
               Falls Übereinstimmung (1-Paar), setze a(2i)
                 und a(2i-1) korrekt und r:=2i.
               Falls keine Übereinstimmung (0-Paar), setze
                 k:=k+1.
               Setze i:=i+1.
             END;
Schritt 2: IF (Bedingung I) THEN
             BEGIN
               Vergleiche den korrekten Wert a(r) mit je einem
                 Vertreter a(i) der 0-Paare.
               Falls Übereinstimmung, setze a(i) korrekt und
                 den anderen Vertreter des 0-Paares inkorrekt.
               Falls keine Übereinstimmung, setze a(i) in-
                 korrekt.
             END
            ELSE
             BEGIN
               Vergleiche den korrekten Wert a(r) mit allen
               noch nicht diagnostizierten Werten a(i).
               Falls Übereinstimmung, setze a(i) korrekt,
                 sonst inkorrekt.
               Stoppe.
             END;
Schritt 3: Setze n:=n-|{diagnostizierte Werte}|,
                t:= t-|{diagnostizierte inkorrekte Werte}|.
           IF (n>0) AND (t>0) THEN
              GOTO Schritt 0
            ELSE
             Setze alle noch nicht diagnostizierten Werte
             korrekt und stoppe.
```

Eine genaue Formulierung der Bedingung I in Schritt 2 ist dem Programm DYNDIA im Anhang zu entnehmen. Sie prüft, ob es vorteilhafter ist, rekursiv im Algorithmus A fortzufahren, oder ob eine

einfache Strategie, auf die Algorithmus B aufgebaut sein wird, die Diagnose schneller vollendet. Man beachte, daß die Indizierung der Werte in Schritt O sich in jedem Durchlauf des Algorithmus ändern kann. Beim Updaten der Parameter n und t in Schritt 3 sind nur die im direkt vorangegangenen Durchlauf der Schritte 1 und 2 diagnostizierten Werte zu berücksichtigen.

Beweis der Richtigkeit von Algorithmus A:

1) Der Algorithmus terminiert, da in jedem Durchlauf in Schritt 2 mindestens ein Wert diagnostiziert wird und Schritt 3 zum Abbruch führt, wenn bereits alle Werte diagnostiziert sind.

2) Nachdem Schritt 1 zum ersten Mal durchlaufen worden ist, ist a(r) stets ein korrekter Wert (da $t \leq (n-1)/2$).
Beweis: Falls $a(2i)=a(2i-1)$ gilt für ein $i \leq \lfloor n/2 \rfloor$, so wird $r=2i$ gesetzt und a(2i) ist sicher ein korrekter Wert. Falls k=t gilt nach Schritt 1, haben sich t O-Paare ergeben, in denen wegen der Annahme von höchstens t inkorrekten Werten jeweils genau ein inkorrekter Wert steckt. Der Wert a(n) gehört zu keinem dieser t Paare und ist deshalb korrekt (man beachte den Anfangswert r=n).

3) Werte werden nur dann inkorrekt gesetzt, wenn sie sich im Vergleich mit a(r) unterscheiden.

4) Abgesehen von a(r), werden Werte nur dann korrekt gesetzt, wenn sie entweder mit einem anderen Wert in Übereinstimmung verglichen werden oder wenn bereits t andere Werte inkorrekt gesetzt sind. q.e.d.

Der nun folgende Satz gibt die maximale Anzahl von Vergleichen an, die Algorithmus A benötigt (worst-case-Betrachtung).

SATZ 5.1.

Seien n Werte gegeben, von denen höchstens t ($t \leq (n-1)/2$) inkorrekt sind und bezeichne $M_A(n,t)$ die maximale Anzahl von Vergleichen, die Algorithmus A benötigt, um diese n Werte dynamisch zu diagnostizieren. Dann gilt:

$$M_A(n,0) = 0 \quad \text{und}$$

$$M_A(n,t) = \begin{cases} \lceil n/2 \rceil + \lceil 3t/2 \rceil - 1 & \text{für n ungerade , t gerade} \\ \lfloor n/2 \rfloor + \lceil 3t/2 \rceil - 1 & \text{sonst} \end{cases}$$

$$\text{für } t>0.$$

Beweis:

Für t=0 wird in Algorithmus A kein Wertepaar verglichen. Sei also t>0. Hier wird der Beweis durch vollständige Induktion über die Zahl t geführt. Für t=1 werden $\lfloor n/2 \rfloor$ Vergleichspaare in Schritt 1 gebildet, falls das letzte dieser Paare ein O-Paar ist und alle anderen Paare 1-Paare. In Schritt 2 wird dann der korrekte Wert a(r) mit einem der Werte im O-Paar verglichen. Also gilt $M_A(n,1) = \lfloor n/2 \rfloor + 1$. Sei nun die Aussage des Satzes bereits für alle Fehlerschranken $\leq t-1$ ($t \geq 2$) gezeigt. Eine Inspektion von Algorithmus A ergibt:

$$\left.\begin{aligned} M_A(n,t) &= n/2 + \max_{1 \leq k \leq t} \min\{2k, k+M_A(k,t-k)\} \text{ für } n \text{ gerade} \\ M_A(n,t) &= \lfloor n/2 \rfloor + \max_{1 \leq k \leq t} \min\{2k+1, k+M_A(k+1,t-k)\} \text{ für } n \text{ ungerade} \end{aligned}\right\} (*).$$

Um die Induktionsvoraussetzung einsetzen zu können, wird eine Unterscheidung in 8 Fälle notwendig (n,t und k können jeweils gerade oder ungerade sein). Hier sei nur der eine Fall durchgerechnet, daß n und t ungerade sind und k=t-1 (k=t ist sicher nicht "worst case"). Falls in (*) $2k+1 \leq \lceil 3t/2 \rceil - 1$ gilt, ist sicher $M_A(n,t) \leq \lfloor n/2 \rfloor + \lceil 3t/2 \rceil - 1$. Sei also nun $2k+1 > \lceil 3t/2 \rceil - 1$ (für $t \geq 3$ sicher möglich), zu zeigen ist dann $k+M_A(k+1,t-k) = \lceil 3t/2 \rceil - 1$. Nach Induktionsvoraussetzung gilt $M_A(k+1,t-k) = \lfloor (k+1)/2 \rfloor + \lceil 3(t-k)/2 \rceil - 1$, da t-k ungerade ist. Außerdem $t-k \leq ((k+1)-1)/2$ wegen $2k+1 > \lceil 3t/2 \rceil - 1$.

Also gilt $k+M_A(k+1,t-k) = k + \lfloor (k+1)/2 \rfloor + \lceil 3(t-k)/2 \rceil - 1$

$= 3k/2 + 1/2 + 3t/2 - 3k/2 - 1$

$= 3t/2 + 1/2 - 1 = \lceil 3t/2 \rceil - 1$, da t ungerade ist. q.e.d.

Anmerkung: Die Bedingung $2k > \lceil 3t/2 \rceil - 1$ bzw. $2k+1 > \lceil 3t/2 \rceil - 1$ ergibt gerade Bedingung I in Algorithmus A.

Da stets $t \leq (n-1)/2$ gilt, ist sicher $M_A(n,t) = O(n)$, d.h. die maximal benötigte Zahl von Vergleichen ist linear in der Zahl der vorliegenden Werte.

Im Gegensatz zur rekursiven Struktur von Algorithmus A ist Algorithmus B sehr einfach aufgebaut.

Algorithmus B:

```
Schritt 0: (Initialisierungen)
           n Anzahl von Werten
           a(1),...,a(n) Menge der vorliegenden Werte
           t maximale Zahl von inkorrekten Werten
           Integervariablen i:=1,k:=0,r:=n.
Schritt 1: (Paarbildung)
           WHILE (i≤⌊n/2⌋) AND (k<t) DO
             BEGIN
               Vergleiche a(2i) und a(2i-1) miteinander.
               Falls Übereinstimmung (1-Paar), setze a(2i)
                 und a(2i-1) korrekt und r:=2i.
               Falls keine Übereinstimmung (0-Paar), setze
                 k:=k+1.
               Setze i:=i+1.
             END;
Schritt 2: IF (k<t) THEN
             BEGIN
               Vergleiche a(r) mit allen noch nicht diagnos-
                 tizierten Werten a(i).
               Falls Übereinstimmung, setze a(i) korrekt,
                 sonst inkorrekt.
             END
            ELSE
             BEGIN
               Vergleiche a(r) mit je einem Vertreter a(i) der
                 0-Paare.
               Falls Übereinstimmung, setze a(i) korrekt und
                 den anderen Vertreter des 0-Paares inkorrekt.
               Falls keine Übereinstimmung, setze a(i) inkor-
                 rekt und den anderen Vertreter des 0-Paares
                 korrekt.
             END;
Schritt 3: Setze alle nicht diagnostizierten Werte korrekt.
```

Die Richtigkeit von Algorithmus B folgt entsprechend der von Algorithmus A. Bezeichnet $M_B(n,t)$ die maximale Zahl von Vergleichen, die Algorithmus B benötigt, um die vorliegenden n Werte zu diagnostizieren, so gilt

$$M_B(n,0) = 0 \ , \quad M_B(n,1) = \lfloor n/2 \rfloor + 1 \text{ und}$$

$$M_B(n,t) = \lceil n/2 \rceil + 2t - 2 \text{ für } t \geq 2.$$

Zum Beweis des letzten Ausdruckes beachte man, daß die maximale Zahl von Vergleichen für k=t-1 benötigt wird (nach Schritt 1).

Während die Algorithmen A und B nur für $0 \le t \le (n-1)/2$ einsetzbar sind, läßt der nun folgende Algorithmus C alle Parameterwerte $0 \le t \le n-2$ zu. Für $t \le (n-1)/2$ ist er jedoch den beiden anderen Algorithmen hinsichtlich der Zahl der Vergleiche unterlegen. Unter einem (k+1)-Block sei nun eine Teilmenge von {a(1),..,a(n)} der Kardinalität k+1 verstanden. Die Restmenge besteht aus den Werten, die sich nicht in einem der (k+1)-Blöcke befinden.

Algorithmus C :

Schritt 0: (Initialisierungen)
n Anzahl der Werte
a(1),...,a(n) Menge der vorliegenden Werte
t maximale Zahl von inkorrekten Werten
Integervariablen k:=0,m:=0,r:=n.

Schritt 1: Setze $k := \lceil t/(n-t-1) \rceil$ und $m := \lfloor n/(k+1) \rfloor$ .
Vergleiche in bis zu m disjunkten (k+1)-Blöcken vollständig (d.h. je 2 Werte miteinander), bis sich eine Übereinstimmung a(i)=a(j) ergibt. In diesem Fall setze r:=i.

Schritt 2: IF (k=0) THEN
Setze alle Werte korrekt und stoppe.

Schritt 3: IF (t>mk) THEN
BEGIN
Vergleiche in der Restmenge vollständig, bis sich eine Übereinstimmung a(i)=a(j) ergibt. Setze dann r:=i.
Vergleiche alle nicht diagnostizierten Werte mit a(r). Falls Übereinstimmung,setze a(i) korrekt, sonst inkorrekt.
END
ELSE
BEGIN
Vergleiche alle nicht diagnostizierten Werte a(i) mit a(r). Falls Übereinstimmung,setze a(i) korrekt, sonst inkorrekt.
END;

Beweis der Richtigkeit von Algorithmus C:

Setze R:=n-m(k+1). Die Parameter k und m werden in Schritt 1 so festgelegt, daß einer der 3 disjunkten Fälle auftritt:

a) In einem der m Blöcke ergibt sich ein Vergleich a(i)=a(j).

b) Fall a) tritt nicht ein und $1 \le R \le 2$. Dann besteht die Restmenge aus korrekten Werten.

c) Fall a) tritt nicht ein und $R \ge 3$. Dann ist t>mk und vollständiges Vergleichen in der Restmenge (Schritt 3) ergibt ein Wertepaar a(i)=a(j).

Stets wird also ein korrekter Wert a(r) gefunden und mit ihm alle undiagnostizierten Werte verglichen. q.e.d.

Es bezeichne $M_C(n,t)$ die maximale Zahl der Vergleiche in Algorithmus C, um n Werte, darunter höchstens t inkorrekte, zu diagnostizieren. Dann gilt

SATZ 5.2.

$$M_C(n,t) = \begin{cases} m' \frac{k(k+1)}{2} + (m-1)(k+1) + R & \text{für } t<mk. \\ m \frac{k(k+1)}{2} + m(k+1) & \text{für } t=mk. \\ m \frac{k(k+1)}{2} + m(k+1) + \frac{R(R-1)}{2} & \text{für } t>mk. \end{cases}$$

Dabei ist $R:=n-m(k+1)$ und $m' := \lfloor t/k \rfloor + 1$.

Beweis:
Vollständiges Vergleichen in einem (k+1)-Block benötigt $k(k+1)/2$ Vergleiche. Im ungünstigsten Fall müssen sämtliche m Blöcke vollständig verglichen werden (falls $t \geq mk$) sowie $m'$ Blöcke (falls $t<mk$).
Gilt $t<mk$, so wurde dann ein 1-Paar $a(i)=a(j)$ gefunden und alle Werte in den anderen m-1 Blöcken und in der Restmenge werden mit dem korrekten Wert verglichen.
Gilt $t=mk$, so haben im ungünstigsten Fall alle Vergleiche in den m Blöcken keine Übereinstimmung gebracht. Die dann vorhandene Restmenge ($R \neq 0$ nach der Wahl von m und k) besteht dann ausschließlich aus korrekten Werten, alle Werte in den m Blöcken müssen mit einem Wert aus der Restmenge verglichen werden.
Gilt schließlich $t>mk$ und haben sich in den Blöcken nur 0-Paare ergeben, so ist hier $R \geq 3$ und in der Restmenge muß vollständig verglichen werden. Anschließend müssen alle Werte in den Blöcken mit einem in der Restmenge gefundenen korrekten Wert verglichen werden. q.e.d.

Für den Rest dieses Abschnitts beschränken wir uns auf eine genauere Untersuchung von Algorithmus B. Zum einen soll geklärt werden, wie gut dieser Algorithmus parallelisierbar ist, d.h. wie viele Testrunden für die vollständige Diagnose nötig sind. Anschließend soll in Abhängigkeit der Wahrscheinlichkeiten, daß

einzelne Werte inkorrekt sind, der Erwartungswert der Zahl der Vergleiche für Algorithmus B berechnet werden.

Zur Parallelisierung von Algorithmus B empfielt sich eine leichte Abänderung des Algorithmus. Es ist nämlich nicht vorteilhaft, daß in Schritt 2 nur ein korrekter Wert a(r) für die weitere Diagnose herangezogen wird. Vielmehr ist es sinnvoll, alle bekannt korrekten Werte parallel zu diesem Zwecke einzusetzen.

SATZ 5.3.
Algorithmus B (leicht abgeändert wie gerade beschrieben) vollendet die dynamische Diagnose von n Werten, darunter höchstens $t \leq (n-1)/2$ inkorrekten, in höchstens s Testrunden. Dabei ist

$$s = \begin{cases} 2 & \text{für } t \in \{0,1\} \\ \lceil 2 \log t \rceil + 2 & \text{für } t \geq 2 \end{cases} .$$

Insgesamt gilt also $s = O(\log t)$.

Beweis:
Schritt 1 benötigt stets 1 Runde. Die Werte von s für $t \leq 1$ können unmittelbar nachgeprüft werden. Sei nun $t \geq 2$.
Fall 1) Nach Schritt 1 ist k=t: Dann steckt in jedem O-Paar genau ein korrekter Wert und n-2t Werte (alle, die nicht zu den Paaren gehören) sind bereits als korrekt erkannt. Um ein O-Paar zu diagnostizieren, genügt ein Vergleich eines Vertreters des Paares mit einem korrekten Wert. Im folgenden werden in jeder Testrunde alle bereits bekannt korrekten Werte eingesetzt, um die verbliebenen O-Paare zu diagnostizieren. Da die Diagnose eines O-Paares stets einen neuen bekannt korrekten Wert ergibt, sind also nach der i-ten Testrunde

$$\sum_{j=0}^{i-2} (n-2t)2^j = (n-2t)(2^{i-1}-1)$$

der O-Paare diagnostiziert. Die Diagnose ist vollständig, wenn $(n-2t)(2^{i-1}-1) \geq t$ gilt. Das bedeutet $2^{i-1}-1 \geq t/(n-2t)$ bzw. $i \geq \left\lceil \log\left(\frac{t}{n-2t}+1\right) \right\rceil + 1$.

Wegen $t/(n-2t) \leq t$ und $\lceil \log(t+1) \rceil \leq \lfloor \log t \rfloor + 1$ genügen dann sicherlich $i = \lfloor \log t \rfloor + 2$ Testrunden. Für $t \geq 2$ gilt aber $\lfloor \log t \rfloor + 2 \leq \lceil 2 \log t \rceil + 2$ .

Fall 2) Nach Schritt 1 ist $1 \leq k \leq t-1$: Es sind $n-2k$ Werte ($n-2k-1$ Werte) als korrekt erkannt, falls $n$ gerade (ungerade) ist. Im folgenden wird der kleinere Wert $n-2k-1$ benutzt. Von den $k$ O-Paaren enthalten höchstens $t-k$ zwei inkorrekte Werte. Um immer eine feste Anzahl von O-Paaren diagnostiziert zu haben (und nicht nur einzelne Werte daraus), wird die benützte Menge von bekannt korrekten Werte immer 2 Testrunden lang beibehalten, bevor neu dazugekommende korrekte Werte mit verwendet werden. Dieses Vorgehen erhöht allenfalls die Zahl der benötigten Testrunden.
Es bezeichne nun $L(t,k,a_2,\ldots,a_i)$ die Anzahl der bekannt korrekten Werte nach der $i$-ten Testrunde ($i$ gerade, die Anfangsrunde von Schritt 1 wird hier nicht mitgezählt), wobei $a_j$ die Anzahl der in den Testrunden $j$ und $j-1$ gefundenen O-Paare mit zwei inkorrekten Werten bezeichnet,

d.h. $\sum_{\substack{j=2 \\ j \text{ gerade}}}^{i} a_j \leq t-k$ .

Dann gilt $L(t,k,a_2,..,a_i) = 2(\ldots(2(2(n-2k-1)-a_2)-a_4)\ldots)-a_i$

und $L(t,k,a_2,..,a_j+1,a_{j+2}-1,..,a_i) < L(t,k,a_2,..,a_j,a_{j+2},..,a_i)$

für $a_{j+2} \geq 1$ und $2 \leq j \leq i-2$. Also haben wir mit $2(t-k) \leq n-2k-1$:

$L(t,k,t-k,0,\ldots,0) = \min L(t,k,a_2,\ldots,a_i)$, wobei das Minimum über alle Tupel $(a_2,\ldots,a_i)$ mit $\sum_{\substack{j=2 \\ j \text{ gerade}}}^{i} a_j \leq t-k$ zu bilden ist.

Im ungünstigsten Fall sind also nach der $i$-ten Testrunde dann die wenigsten korrekten Werte bekannt, falls in den ersten beiden Runden nach Schritt 1 bereits alle O-Paare mit zwei inkorrekten Werten diagnostiziert werden. Es gilt

$$L(t,k,t-k,0,\ldots,0) = \left[2(n-2k-1)-(t-k)\right] 2^{i/2-1}$$
$$= (2n-3k-t-2)\,2^{i/2-1} .$$

Die Anfangsrunde hinzugezählt, reichen also $i+3$ Runden sicher aus, falls $(2n-3k-t-2)\,2^{i/2-1} \geq k+1$ gilt (Nach der $(i+1)$-ten Runde liegen dann $k+1$ bekannt korrekte Werte vor, die in den beiden letzten Runden die Diagnose sicher vollenden. Statt $k$ wurde hier die Schranke $k+1$ gewählt, um für ungerades $n$ den isolierten Wert $a(n)$ zu erfassen).

Dies bedeutet $i \geq 2 \log \frac{k+1}{2n-3k-t-2} + 1$ (*).

Es ist $2n-3k-t-2 \geq 2n-4t+1 = 2(n-2t)+1 \geq 3$ (wegen $k \leq t-1$).
Eine Vergrößerung der rechten Seite von (*) erhöht allenfalls die Zahl der benötigten Testrunden. Es ist

$$2 \log \frac{k+1}{2n-3k-t-2} + 1 \leq 2 \log(t/3)+1 = 2 \log t - 2 \log 3 + 1 < 2 \log t - 1.$$

Also genügen $\lceil 2 \log t \rceil + 2$ Testrunden . q.e.d.

Im Algorithmus B sind höchstens $\lceil n/2 \rceil + 2t - 2$ Vergleiche durchzuführen, bis die Diagnose vollständig ist. Interessant ist auch die Angabe, wie viele Vergleiche im Mittel nötig sind. Um einen solchen Erwartungswert auszurechnen, muß eine Annahme über die Wahrscheinlichkeiten gemacht werden, daß einzelne Werte inkorrekt sind. Bezeichnet E den gesuchten Erwartungswert und P(j) die Wahrscheinlichkeit, daß durch Algorithmus B genau j Vergleiche durchzuführen sind, so gilt

$$E = \sum_{j \in \mathbb{N}} j\, P(j) = \sum_{j=0}^{2n} j\, P(j) \quad \text{(da im worst case } \lceil n/2 \rceil + 2t - 2 \leq 2n \text{ Vergleiche durchzuführen sind)}$$
$$= \sum_{j=0}^{2n} j \sum_{i=0}^{t} P(j|i)\, P^*(i)$$

nach dem Gesetz von der totalen Wahrscheinlichkeit, wobei
$P^*(i) = P\{\text{es liegen genau i inkorrekte Werte vor}\}$ und
$P(j|i) = P\{\text{j Vergleiche sind nötig} | \text{es liegen genau i inkorrekte Werte vor}\}$ ist.

Also $E = \sum_{i=0}^{t} P^*(i) \sum_{j=0}^{2n} j\, P(j|i)$ .

Die Ausdrücke $E(i) := \sum_{j=0}^{2n} j\, P(j|i)$ $(0 \leq i \leq t)$ gilt es im folgenden zu berechnen ( E(i) ist ein bedingter Erwartungswert, nämlich der Erwartungswert der Zahl der Vergleiche unter der Bedingung, daß genau i Werte ($i \leq t$) tatsächlich inkorrekt sind).

SATZ 5.4.

Seien $n \geq 3$ und $0 \leq t \leq (n-1)/2$ vorgegeben und es werde Algorithmus B zur dynamischen Diagnose angewendet. Dann gilt:

Für $0 \leq i \leq t-1$:

$$E(i) = \begin{cases} \frac{n}{2} + \sum_{k=\lceil \frac{i}{2} \rceil}^{i} 2k \, \frac{\binom{n/2}{k}\binom{k}{i-k}2^{2k-i}}{\binom{n}{i}} & \text{für } n \text{ gerade} \\ \lceil \frac{n}{2} \rceil + \sum_{k=\lceil \frac{i}{2} \rceil}^{i} 2k \, \frac{\binom{\lfloor n/2 \rfloor}{k}\binom{k}{i-k}2^{2k-i}}{\binom{n}{i}} + \sum_{k=\lceil \frac{i}{2} \rceil}^{i-1} 2k \, \frac{\binom{\lfloor n/2 \rfloor}{k}\binom{k}{i-1-k}2^{2k-i+1}}{\binom{n}{i}} & \text{für } n \text{ ungerade.} \end{cases}$$

Für $i=t$:

$$E(t) = \begin{cases} \binom{n}{t}^{-1}\left[\sum_{k=\lceil \frac{t}{2} \rceil}^{t-1}\binom{n/2}{k}\binom{k}{t-k}2^{2k-t}(\frac{n}{2}+2k) + 2^t\sum_{j=t}^{n/2}\binom{j-1}{j-t}(j+t)\right] & \text{für } n \text{ gerade} \\ \binom{n}{t}^{-1}\left[\sum_{k=\lceil \frac{t}{2} \rceil}^{t-1}\binom{\lfloor n/2 \rfloor}{k}\binom{k}{t-k}2^{2k-t}(\lfloor \frac{n}{2} \rfloor +2k+1) + \right. \\ \left. \sum_{k=\lceil \frac{t}{2} \rceil}^{t-1}\binom{\lfloor n/2 \rfloor}{k}\binom{k}{t-1-k}2^{2k+1-t}(\lfloor \frac{n}{2} \rfloor +2k+1) + 2^t\sum_{j=t}^{\lfloor n/2 \rfloor}\binom{j-1}{j-t}(j+t)\right] & \text{für } n \text{ ungerade .} \end{cases}$$

Beweis:

Fall $0 \leq i \leq t-1$: Die Werte $n/2+2k$ bzw. $\lceil n/2 \rceil + 2k = \lfloor n/2 \rfloor + 2k+1$ ($n$ gerade bzw. ungerade) geben die jeweils nötige Zahl der Vergleiche an. Die Summationsvariable $k$ entspricht der Zahl der O-Paare in Schritt 1 von Algorithmus B. Für gerades $n$ sind $k$ der $n/2$ Paare O-Paare ($\binom{n/2}{k}$ Möglichkeiten), davon enthalten $i-k$ viele zwei inkorrekte Werte ($\binom{k}{i-k}$ Möglichkeiten), während die restlichen $2k-i$ O-Paare genau einen korrekten Wert enthalten (jeweils 2 Möglichkeiten $\hat{=}\ 2^{2k-i}$ Möglichkeiten). Von allen $\binom{n}{i}$ Fällen, daß genau $i$ inkorrekte Werte vorliegen, ergeben also $\binom{n/2}{k}\binom{k}{i-k}2^{2k-i}$ viele gerade $k$ O-Paare. Für ungerades $n$ muß außerdem noch unterschieden werden, ob der $n$-te Wert, der nicht in die Paarbildung von Schritt 1 einbezogen ist, korrekt oder inkorrekt ist. Im ersten Summand ist er korrekt, im zweiten Summanden inkorrekt.

Fall i=t: Schritt 1 in Algorithmus B wird abgebrochen, falls sich bereits t O-Paare ergeben haben. Falls sich k<t O-Paare ergeben, stimmen die Formelanteile mit denen für i<t überein. Falls t O-Paare vorliegen, gibt es $2^t$ Möglichkeiten für eine Verteilung der t inkorrekten Werte in diesen t Paaren (genau einer davon in jedem Paar). Es sind gerade j+t Vergleiche nötig, wenn nach Bildung von j Paaren diese t O-Paare gefunden wurden. In diesem Fall gibt es $\binom{j-1}{j-t}$ Möglichkeiten für die j-t 1-Paare, sich unter den ersten untersuchten j-1 Paaren zu befinden (das letzte gebildete Paar war natürlich ein O-Paar). q.e.d.

Zur vollständigen Berechnung des Erwartungswertes E müssen jetzt noch Annahmen über die Wahrscheinlichkeiten $P^*(i)$ $(0 \le i \le t)$ gemacht werden. Als eine Möglichkeit dafür bietet sich an, eine Wahrscheinlichkeit $0 \le p < 1$ anzunehmen, daß ein einzelner Wert zur Zeit der Diagnose inkorrekt ist. Die Werte $P^*(i)$ berechnen sich dann bei statistischer Unabhängigkeit zu

$$P^*(i) = \binom{n}{i} p^i (1-p)^{n-i} \Big/ \sum_{s=0}^{t} \binom{n}{s} p^s (1-p)^{n-s} .$$

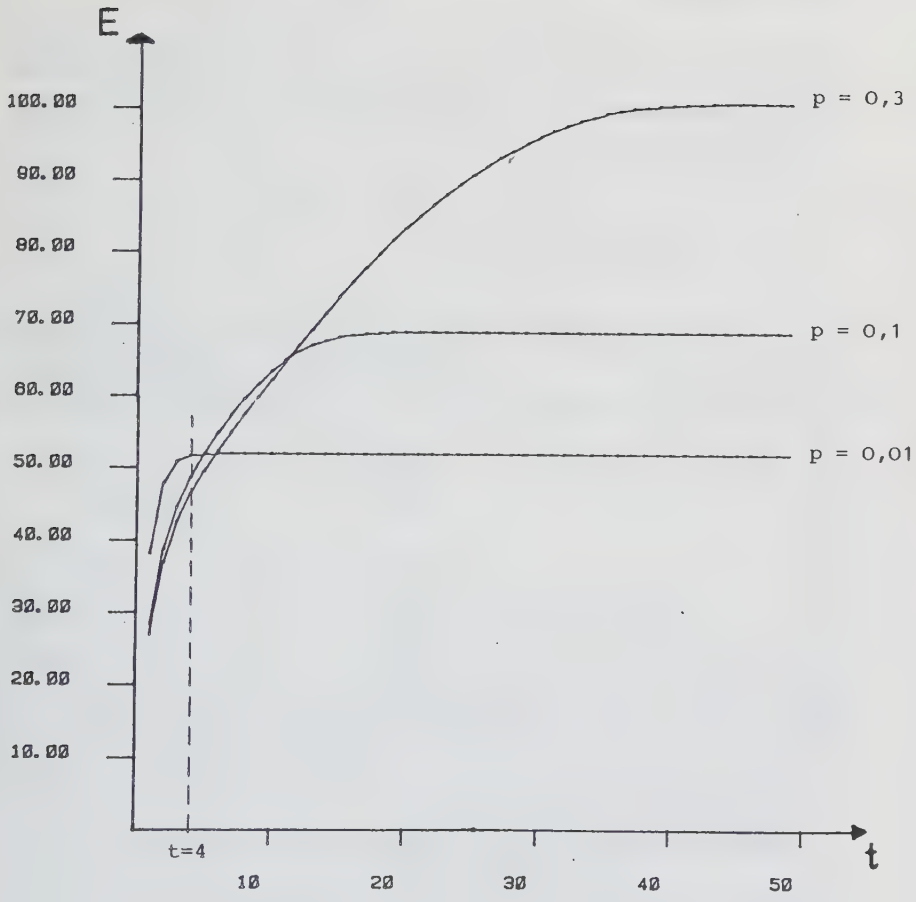
Der Nenner ergibt sich dabei aus der Annahme, daß nicht mehr als t inkorrekte Werte vorliegen dürfen.

Damit liegen nun alle Angaben vor, um für gegebene Parameter $n \ge 3$ und $0 \le t \le (n-1)/2$ den Erwartungswert E für die Zahl der Vergleiche in Algorithmus B auszurechnen, nämlich

$$E = \sum_{i=0}^{t} P^*(i)\ E(i).$$

Anmerkung: Algorithmus A ist so aufgebaut, daß er in jedem einzelnen vorliegenden Diagnosefall höchstens so viele Vergleiche benötigt wie Algorithmus B. Der Erwartungswert E läßt sich deshalb auch als obere Schranke für den Erwartungswert der Zahl der Vergleiche in Algorithmus A deuten.

Für die feste Anzahl von n=100 vorliegenden Werten zeigt Figur 5.1. den Verlauf von E in Abhängigkeit von t. Dabei sind 3 Kurven für p=0.01, p=0.1 und p=0.3 eingezeichnet.




Figur 5.1. E-Werte für Algorithmus B

Interessant an diesen Kurven ist sicherlich, daß zwar für größere Werte von t auch der Wert E für wachsendes p zunimmt, daß für kleine Werte von t aber gerade das entgegengesetzte Verhalten zu beobachten ist ( man betrachte die in Figur 5.1. eingezeichnete Stelle t=4). Dies findet eine Erklärung darin, daß bei kleinem t und wachsendem p die Wahrscheinlichkeit, daß k=t O-Paare gefunden werden, stark zunimmt und daß in diesen Fällen

nicht sämtliche n/2 bzw. $\lfloor n/2 \rfloor$ Paare in Schritt 1 gebildet werden müssen.

Zum Abschluß dieses Abschnitts hier noch einige Zahlenbeispiele, um den Unterschied zwischen worst case und durchschnittlichem Fall zu veranschaulichen (Figur 5.2.). Stets wird dabei n=100 und p=0.1 angenommen.

| t | Zahl der Vergleiche in Algorithmus B ( worst case ) | Zahl der Vergleiche in Algorithmus B ( Erwartungswert ) | Zahl der Vergleiche in Algorithmus A ( worst case ) |
|---|---|---|---|
| 1 | 51 | 28,44 | 51 |
| 5 | 58 | 52,04 | 57 |
| 10 | 68 | 61,69 | 64 |
| 15 | 78 | 68,32 | 72 |
| 20 | 88 | 68,98 | 79 |
| 23 | 94 | 68,99 | 84 |

Figur 5.2. Vergleich zwischen worst-case- und Erwartungswerten.

## 5.2. EINE GEGENÜBERSTELLUNG VON STATISCHER UND DYNAMISCHER DIAGNOSE

In Abschnitt 4.3. wurde das Vergleichstestmodell VT2 in seiner statischen Erscheinungsform untersucht, der vorangegangene Abschnitt 5.1. dagegen konzentrierte sich auf die dynamische Diagnose in diesem Modell. Hier soll nun eine Gegenüberstellung dieser beiden Organisationsformen der Diagnose vorgenommen werden, gestützt auf die folgenden Unterscheidungsmerkmale:

a) Festlegung der Diagnosestruktur: Zeitpunkt und in Abhängigkeit von welchen Größen.
   Technische Voraussetzungen.
b) Ablauf von Test- und Diagnosephase.
c) Verwaltungsaufwand.
d) Flexibilität hinsichtlich der Organisation der Diagnose.
e) Komplexitätsbetrachtungen: Zahl der Vergleiche und der Testrunden.

a) Festlegung der Diagnosestruktur:
Bei der statischen Diagnose wird die Diagnosestruktur zu Beginn vollständig festgelegt in Abhängigkeit der Parameter n und t. Normalerweise wird man dabei einen t-optimalen Diagnosegraphen mit n Knoten wählen. Wichtiges Kennzeichen ist, daß der Diagnosegraph unabhängig von der konkreten Fehlersituation gewählt wird. Dagegen wird in der dynamischen Diagnose die Diagnosestruktur erst während der Diagnose aufgebaut, abhängig von schon gewonnenen Diagnoseinformationen und von der vorliegenden Fehlersituation.
Im Prinzip muß hier im Gegensatz zur statischen Diagnose jedes Paar von Werten miteinander verglichen werden können. Werden also die Werte durch Ergebnisse von Prozessoren in einem Multi-Prozessorsystem dargestellt, so müssen die entsprechenden Kommunikationsverbindungen zwischen je zwei Prozessoren vorhanden sein (z.B. globaler Bus oder vollständige Vernetzung oder Übertragung über Vermittlerprozessoren (vgl. [65])). Eine genaue Inspektion von Algorithmus A (bzw. seiner Realisierung durch das Programm DYNDIA im Anhang) zeigt allerdings, daß auch hier nicht alle Vergleiche von Wertepaaren möglich sein müssen. Die Wertepaare mit den Indexpaaren (i,j) mit $i,j \leq n$ und i,j ungerade werden niemals verglichen.

b) Ablauf von Test- und Diagnosephase:
Test- und Diagnosephase sind in der statischen Organisation zeitlich und logisch getrennt. Nach Aufbau des Diagnosegraphen und nach Durchführung aller in ihm vorgesehener Tests beginnt erst die Diagnosephase. Bei der dynamischen Diagnose sind die beiden Phasen verschmolzen. Nach der Durchführung jedes einzelnen Tests wird, soweit möglich, diagnostiziert und in Abhängigkeit davon der nächste durchzuführende Test festgelegt.

c) Verwaltungsaufwand:
Statische Diagnose: In Abhängigkeit von n und t muß die Diagnosestruktur festgelegt (evtl. nur Nachschauen in einer abgespeicherten Liste von t-optimalen Diagnosegraphen) und die Durchführung aller dieser Tests veranlaßt werden. Schließlich muß die Diagnoseinstanz mit Hilfe eines Diagnosealgorithmus die Diagnose vornehmen.
Dynamische Diagnose: Abhängig von n und t, läßt die Diagnoseinstanz etwa einen der Algorithmen A, B oder C durchlaufen.

d) Flexibilität der Organisation:
Die dynamische Diagnose verlangt eine globale Organisation von Testeinsatz und Diagnose, da bei der Festlegung jedes einzelnen Tests und bei jeder Diagnosetätigkeit sämtliche bis dahin gewonnene Information verwendet wird. Dagegen kann die statische Diagnose zentral oder dezentral lokal organisiert sein.

e) Komplexitätsbetrachtungen:
Bei der statischen Diagnose benötigen t-optimale Diagnosegraphen mit n Knoten $\lceil 2n/3 \rceil$ Vergleiche für t=1 (bei nicht zusammenhängendem Graphen) , $\lceil nt/2 \rceil$ Vergleiche für $2 \leq t \leq (n-1)/2$. Diese Vergleiche können in $t \leq l \leq t+2$ Testrunden organisiert werden (vgl. die Abschnitte 4.3. und 4.5.).
Für die dynamische Diagnose soll hier Algorithmus B herangezogen werden. Er benötigt $\lfloor n/2 \rfloor + 1$ Vergleiche für t=1 bzw. $\lceil n/2 \rceil + 2t - 2$ Vergleiche für $2 \leq t \leq (n-1)/2$ im schlechtesten Fall. Wie Figur 5.2. in Abschnitt 5.1. gezeigt hat, liegt der Erwartungswert der Zahl der Vergleiche noch beträchtlich niedriger. Algorithmus B benötigt hierzu höchstens $\lceil 2 \log t \rceil + 2$ Testrunden.

Betrachtet man nur die Komplexitätsklassen dieser Angaben, so zeigt sich die Überlegenheit der dynamischen Diagnose (mit Algorithmus B, für $t \leq (n-1)/2$) sehr deutlich (Figur 5.3.).

| | statische Diagnose | dynamische Diagnose |
|---|---|---|
| Zahl der Vergleiche | $O(n^2)$ | $O(n)$ |
| Zahl der Testrunden | $O(t)$ | $O(\log t)$ |

Figur 5.3.

Das Verhältnis der Zahl der Vergleiche zwischen dynamischer und statischer Diagnose (worst-case-Werte für die dynamische Diagnose) ergibt sich zu

$$\frac{\left\lfloor \frac{n}{2} \right\rfloor + 1}{\left\lceil \frac{2n}{3} \right\rceil} \approx \frac{3}{4} \quad \text{(n groß)} \quad \text{für } t=1$$

und

$$\frac{\left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{3t}{2} \right\rceil - 1}{\left\lceil \frac{nt}{2} \right\rceil} \approx \frac{1}{t} + \frac{3}{n} - \frac{2}{nt} \longrightarrow 0 \text{ (n groß) für } 2 \leq t \leq (n-1)/2,$$

wenn hier Algorithmus A als Vertreter der dynamischen Diagnose herangezogen wird. Insbesondere für große Werte von n bringt die dynamische Diagnose eine gewaltige Einsparung an Testaufwand.

Zusammenfassend ist also ein Vorteil der statischen Diagnose die gegebene Flexibilität in der Organisation von Testeinsatz und

Diagnose. Die dynamische Diagnose muß global organisiert sein, hat aber dafür enorme Vorteile hinsichtlich Verwaltungsaufwand und vor allem hinsichtlich der Zahl der durchzuführenden Vergleiche (zumindest für $t \leq (n-1)/2$). Hier zur Veranschaulichung der letzten Aussagen noch einige Zahlenwerte (Figur 5.4.).

| Parameter n= | Parameter t= | dynamische Diagnose ( Algorithmus A ) | statische Diagnose |
|---|---|---|---|
| 10 | 4 | 10 | 20 |
| 20 | 9 | 23 | 90 |
| 50 | 20 | 54 | 500 |
| 100 | 40 | 109 | 2000 |
| 1000 | 400 | 1099 | 200000 |

Figur 5.4. Zahl der Vergleiche bei dynamischer und statischer Diagnose.

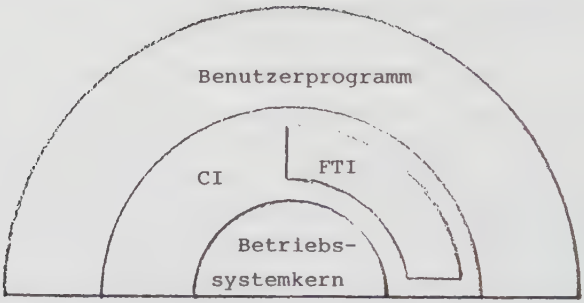
# 6. ATTEMPTO : EIN KONZEPT ZUR REALISIERUNG EINES SELBSTDIAGNOSE-MODELLS

ATTEMPTO ist ein fehlertolerantes Multiprozessor-System, das gegenwärtig am Institut für Informationsverarbeitung der Universität Tübingen entwickelt wird. Meine Mitarbeit an diesem Projekt konzentriert sich auf die theoretischen Aspekte von Test und Diagnose und auf Konzepte ihrer Verwirklichung in ATTEMPTO (darüber mehr in den Abschnitten 6.1. und 6.2.). Hier zuächst einige allgemeine Informationen über das System, detailliertere Angaben finden sich in [9].

Vom Standpunkt des Benutzers aus ist ATTEMPTO ein "Single User-Multitasking"-System, seine Realisierung als Multiprozessor-System bleibt ihm verborgen. Für jeden einzelnen Job kann er einen sogenannten Toleranzgrad t wählen. Damit wird ihm garantiert, daß er einen korrekten Output für seinen Job bekommt unter der Voraussetzung, daß während der Abarbeitung des Jobs Fehler in höchstens t Prozessorplatinen aufgetreten sind (der ebenfalls wählbare Wert t=-1 bewirkt eine wenigstens fehlererkennende Art der Abarbeitung). Da jeder Job mit Toleranzgrad $t \geq 1$ unabhängig auf mehreren Prozessoren bearbeitet wird, sinkt natürlich mit wachsendem t der Systemdurchsatz.

ATTEMPTO besteht aus einer Reihe von Single-Board-Computern (SBCs), die über einen multiprozessorfähigen Bus miteinander kommunizieren können. Die Systemsoftware baut sich aus identischen, lokalen und autonomen Betriebssystemen (ATOS) auf - eines auf jedem SBC. Den Kern von ATOS bildet ein konventionelles Mikroprozessor-Betriebssystem, das Fehlertoleranzkonzept wird durch den Einbau einer Zwischenschicht zwischen Benutzerjob und diesem Betriebssystem verwirklicht (Figur 6.1.).

Dabei ist die Kommunikationsinstanz (CI) für die Kommunikation in ATTEMPTO verantwortlich, die Fehlertoleranzinstanz (FTI) verwirklicht die Fehlertoleranzanforderungen. Diese Zwischenschicht ist transparent für den Benutzer und schützt damit




Figur 6.1. Die Softwareschichten von ATOS

den Systemkern vor unberechtigten Zugriffen. Die Kommunikation zwischen verschiedenen SBCs dient ausschließlich Fehlertoleranzzwecken, wird über die jeweiligen CIs abgewickelt und benutzt "Messages".

## 6.1. DIE EINORDNUNG VON ATTEMPTO IN DAS SELBSTDIAGNOSE-PARAMETERSYSTEM

Tests:

2.1.1. Beschaffenheit der Einheiten des Systems:
Single-Board-Computer, Ausfälle sind stochastisch unabhängig. Da identische Einheiten gewählt werden, fallen sie auch mit gleicher Wahrscheinlichkeit aus.

2.1.2. Art der zugelassenen Fehler:
Permanente und transiente Fehler

2.1.3. Beschaffenheit der Tests:
Disjunkte 1-1-Tests, m=2, vollständige Tests (jeder Fehler, der das Programmergebnis beeinflußt, wird erfaßt), ungerichtete Tests (Vergleich von Ergebnissen).

2.1.4. Organisation des Testeinsatzes:
Im Normalfall funktionsbegleitend. Sporadisches Testen auf Anweisung des Benutzers.

Diagnose:

2.2.1. Voraussetzungen der Diagnose:
t-Fehlerannahme (t=Toleranzgrad des Jobs), intakte Einheiten dürfen nicht ersetzt werden, 1-Schritt-diagnose, es gelten die Annahmen von VT2 (vgl. 4.3.).

2.2.2. Festlegung der Testmenge:
statisch (Verwendung von t-optimalen Diagnosegraphen), keine Mehrfachanwendung von Vergleichstests.

2.2.3. Organisation der Auswertung der Testergebnisse:
dezentral, lokal und nicht hierarchisch.

## 6.2. DAS FEHLERTOLERANZKONZEPT VON ATTEMPTO

a) Testen in ATTEMPTO:

Grundsätzlich besteht ein Test in ATTEMPTO aus einem Vergleich von Ergebnissen, die zwei Prozessoren nach Bearbeitung von identischen Kopien eines Programms errechnet haben. Um permanente und transiente Fehler unterscheiden zu können, sind zusätzlich zu diesen Vergleichstests noch Prozessor-Selbsttests vorgesehen. Falls ein Vergleichstest einen Fehler einer Einheit erkannt hat und der anschließende Selbsttest der Einheit keinen Fehler findet, wird auf einen transienten Fehler geschlossen (falls dies nicht zu häufig geschieht). Findet jedoch der Selbsttest einen Fehler, so ist dieser permanent (diese Selbsttests ergänzen die Vergleichstests, sie sind in 6.1. nicht eingefügt).

Als Einheiten der Fehlerlokalisierung werden die SBCs des Systems angesehen, bei Auftreten eines Fehlers wird die Platine als ganze als fehlerhaft angesehen. Gegen ein Versagen im Systembus besteht in der ersten Version von ATTEMPTO keine Vorkehrung. Allgemein werden Tests bei folgenden Gelegenheiten eingesetzt:

- Start-up-Tests: Um bei Betriebsbeginn ein voll funktionsfähiges System vorzufinden, werden Eingangstests gestartet.
- Funktionsbegleitende Vergleiche von Jobergebnissen sowie anschließende Selbsttests.
- "freies Testen": Jeder Prozessor, der gerade nicht mit der Abarbeitung eines Benutzerjobs beschäftigt ist, durchläuft einen Selbsttest.
- Tests zur Systemrekonfiguration: Das gesamte System wird mit Vergleichs- und Selbsttests geprüft.

Als Daten für die in der Diagnosephase durchzuführenden Vergleichstests werden nun nicht die vollständigen Outputdaten des betreffenden Jobs herangezogen, sondern nur eine komprimierte Version der Bitlänge r=16 ( der Wert 16 rührt dabei von der Bitbreite des verwendeten Systembus her). Dadurch wird verhindert, daß der Bus durch die Kommunikation der manchmal sehr umfangreichen Outputdaten (etwa lange Ergebnistabellen) zu Diagnosezwecken überlastet wird. Für die Datenkompression sind im wesentlichen zwei Verfahren bekannt.
Im ersten Verfahren ist eine soganannte Zählfunktion vorgegeben, die auf die vorgegebenen Daten angewendet wird [33]. Meist berechnen diese Zählfunktionen die Anzahl von 0-1-Übergängen oder einfach die Anzahl der 1-Bits in der Datenfolge (weitere Zählfunktionen sind in [33] aufgelistet). Daraus folgt, daß die Länge der komprimierten Daten etwa den (binären) Logarithmus der Länge der vorgegebenen Daten beträgt.
Eine Kompression auf eine feste vorgegebene Bitlänge r ermöglicht das zweite Verfahren. Hier durchläuft die Datenfolge ein sogenanntes lineares Schieberegister mit Feedback (LFSR) [15,43,48,82], ein sogenanntes Signaturregister. In [15] werden Implementierungsprobleme auf verschiedenen Ebenen betrachtet, in [43] werden Signaturregister mit seriellen und parallelen Eingängen gegenübergestellt, [48] schließlich berechnet Fehlererkennungswahrscheinlichkeiten von verschiedenen Registern. In ATTEMPTO wird eine Softwareversion eines LFSR verwirklicht (vgl. [82]). Ein Datenstrom der Länge k, dessen Bits als Koeffizienten eines binären Polynoms vom Grad k-1 aufgefaßt werden, wird durch ein fest vorgegebenes binäres Polynom q vom Grad r (in ATTEMPTO r=16) geteilt, q heißt charakteristisches Polynom. Der entstehende Divisionsrest ist die komprimierte Version des Datenstroms, die Signatur. Nach Theorem 3 in [82] führen zwei Datenströme, die sich in genau einem Bit unterscheiden, dann stets zu verschiedenen Signaturen (falls q mindestens zwei von 0 verschiedene Koeffizienten hat). Das bedeutet, daß alle Einzelbitfehler zu einer Veränderung der Signatur führen und deshalb auch nach der Datenkompression erkennbar bleiben.

Bezeichnet nun P die Wahrscheinlichkeit dafür, daß eine korrekte und eine fehlerhafte Datenmenge identische Signaturen

ergeben, und nimmt man an, daß alle möglichen Abweichungen eines Datenstromes gleichwahrscheinlich sind, dann gilt [82]:

$$P = \frac{2^{k-r} - 1}{2^k - 1} \qquad (k \geq r).$$

In unserem Fall (r=16) nähert sich dieser Wert P dem Wert $2^{-16} \approx 0.000015$, falls $k \to \infty$ geht.

b) Diagnose in ATTEMPTO:
Die formalen Parameter der Diagnose in ATTEMPTO sind in Abschnitt 6.1. aufgeführt. Um den Systembus nicht unnötig zu belasten und keine unnötigen Vergleiche von Signaturen durchführen zu müssen, wird in Abhängigkeit von der vorgegebenen Fehlerschranke t (dem Toleranzgrad des Jobs) eine minimale Menge von Vergleichen festgelegt, um t-Diagnostizierbarkeit zu garantieren (Datenselektion). Diagnosegraphen, die t-optimal in Modell VT2 sind, leisten hier das gewünschte. Dabei ist $|N|=1$ für $t=0$, $|N|=3$ für $t=1$ und $|N|=t+3$ für $t \geq 2$ (für $t=-1$ wird nur Fehlererkennung angestrebt. Dieser Fall wird hier nicht weiter betrachtet), die betreffenden Diagnosegraphen können Figur 4.5. entnommen werden. Die Knotenzahl $|N|=t+3$ (für $t \geq 2$) wird dabei gewählt, um für eine überwachte Outputphase des Jobs (Teil c) dieses Abschnitts) mindestens 3 fehlerfreie Prozessoren vorliegen zu haben. Als Knoten für diese Diagnosegraphen dienen die am Job beteiligten Prozessoren bzw. ihre Signaturen, die Signaturen zweier Prozessoren werden verglichen, wenn eine Kante zwischen den beiden entsprechenden Knoten besteht.

Dies ist die Gesamtsicht der Diagnose, in ATTEMPTO ist aber eine dezentrale und lokale Diagnose realisiert. Jeder am Job beteiligte Prozessor sieht nur einen begrenzten Ausschnitt des benutzten Diagnosegraphen. In Abhängigkeit von t berechnet er seine Nachbarmenge von Prozessoren in dem zu bildenden Graphen, d.h. die Menge der Prozessoren, deren Signaturen er mit seiner Signatur vergleicht. Nimmt man eine Numerierung der am Job beteiligten Prozessoren von 1 bis t+3 an (für $t \geq 2$), so vergleicht der i-te Prozessor bei geradem t etwa seine Signatur mit den Signaturen der Prozessoren mit den Nummern

$i+1, i+2, \ldots, i+t/2, i-1, i-2, \ldots, i-t/2$ (modulo $t+3$).
Man beachte im folgenden, daß für $t \geq 2$ alle benutzten Diagnosegraphen sogar stark t-optimal sind (vgl. Abschnitt 4.3.).

Jeder Prozessor führt nun folgenden Diagnosealgorithmus durch. Zunächst vergleicht er seine eigene Signatur mit der Signatur jedes seiner Nachbarprozessoren im Diagnosegraphen. Falls wenigstens einer dieser Signaturvergleiche eine Übereinstimmung ergibt, kann er sicher sein, die korrekte Signatur berechnet zu haben (wegen der Annahmen von Modell VT2) und kann alle seine Nachbarn korrekt diagnostizieren. Ein fehlerhafter Prozessor aber (,der eine fehlerhafte Signatur berechnet hat,) wird keinen Nachbarn mit gleicher Signatur finden. Dennoch kann er sich natürlich irrtümlicherweise für fehlerfrei halten und alle seine Nachbarprozessoren für fehlerhaft.

Bei Auftreten von genau t fehlerhaften Prozessoren kann mit geringer Wahrscheinlichkeit der Fall eintreten, daß die Nachbarmenge eines fehlerfreien Prozessors gerade aus der Menge der fehlerhaften Prozessoren besteht und so dieser Prozessor keine mit seiner übereinstimmende Signatur bei seinen Nachbarn findet ($t \geq 2$). In diesem Fall haben sich aber 2 andere fehlerfreie Prozessoren verglichen (starke t-Diagnostizierbarkeit des Diagnosegraphen), stellen fest, daß sie nur jeweils einen ihrer Nachbarn für fehlerfrei halten (ein weiterer Kommunikationsschritt) und melden dann diesem isolierten fehlerfreien Prozessor, daß er fehlerfrei ist. Die entsprechende Situation für $t=1$ liegt vor, falls der Prozessor mit Nummer 2 ( der mittlere in der 3-Kette) fehlerhaft ist. In dieser Situation kann sich jeder der 3 Prozessoren für fehlerfrei halten.

Nun haben alle fehlerfreien Prozessoren die Gewißheit, daß sie fehlerfrei sind, und die korrekte Diagnosesicht über ihre Nachbarprozessoren. Über die Diagnosesicht fehlerhafter Prozessoren läßt sich keine Aussage machen. Jeder Prozessor führt eine Fehlerhäufigkeitsliste über alle Prozessoren im System, in der eingetragen wird, wie oft jeder Prozessor von ihm für fehlerhaft angesehen wurde. Übersteigt diese Zahl eine vorgegebene Schranke, wird jede weitere Zusammenarbeit mit diesem Prozessor verweigert, da die aufgetretenen Unstimmigkeiten

dann offensichtlich nicht mehr durch transiente Fehler zu erklären sind.

c) Bestimmung des Output-Prozessors und Überwachung des Outputs:
Da es in ATTEMPTO keine zentrale globale Diagnosesicht für das System gibt und da die Produktion von korrekten Daten das wichtigste Entwurfsziel des Systems ist und da nach der Diagnosephase b) sich auch defekte Prozessoren für fehlerfrei halten können und den Output des Jobs übernehmen könnten, ist noch ein zusätzlicher Mechanismus notwendig, um zu garantieren, daß der Output der Ergebnisse des Jobs nur von einem fehlerfreien Prozessor stammen kann und korrekt ist.
Dazu bedienen wir uns kryptographischer Methoden [26,27]. Hier soll zuerst die grundsätzliche Methodik des Vorgehens beschrieben werden.
Um den Output von Daten beginnen zu können, benötigt jeder beteiligte Prozessor einen Schlüssel S (zum Beispiel die Startaddresse für die WRITE-Routine). Dieser Schlüssel muß ihm von einem anderen Prozessor zur Verfügung gestellt werden. Prozessor i fragt einen der Prozessoren j, die er für fehlerfrei hält, nach diesem Schlüssel S. Prozessor j schickt S an Prozessor i zurück, verschlüsselt in Abhängigkeit von j's Signatur. Prozessor i kann aus dieser Mitteilung S nur dann herauslesen, wenn seine Signatur mit der von Prozessor j übereinstimmt. Fehlerhafte Prozessoren aber finden keine anderen Prozessoren mit gleicher Signatur und deshalb auch niemals den benötigten Schlüssel S.
Hier ein Beispiel einer Realisierung dieser Methode:

S sei die Startaddresse der WRITE-Routine und wie die Signatur ein 16-Bit-Wort. Sign(x) bezeichne die von Prozessor x berechnete Signatur.
Prozessor j schickt die (Boolsche) Summe S ⊕ Sign(j) an Prozessor i zurück, der S daraus durch eine weitere (Boolsche) Addition (S ⊕ Sign(j)) ⊕ Sign(i) berechnet. Nur für Sign(i)=Sign(j) (wenn also Prozessor i fehlerfrei ist) gilt (S ⊕ Sign(j)) ⊕ Sign(i) = S und Prozessor i kann die WRITE-Routine aufrufen.

Deshalb können nur fehlerfreie Prozessoren den Output durch-

führen. Dies geschieht in einer Weise, daß einer der fehlerfreien Prozessoren seine Daten ausgibt und 2 andere fehlerfreie Prozessoren (für $t \geq 2$) diesen Output durch Vergleich mit den ihnen vorliegenden Outputdaten überwachen. Dies bedeutet eine Absicherung gegen Fehler während der Outputoperation (zumindest eine Fehlererkennung).

## 6.3. ANMERKUNGEN ZU ATTEMPTO

a) Das beschriebene Diagnoseverfahren in ATTEMPTO erscheint sehr aufwendig, wenn man etwa als Alternative eine Majoritätsentscheidung unter der Signaturen betrachtet. Ein solches Verfahren würde aber eine etwas höhere Busbelastung mit sich bringen (jeder beteiligte Prozessor benötigt anstatt t nun alle Signaturen) und vor allem eine viel höhere Redundanz bedeuten, um t Fehler tolerieren zu können (ein Toleranzgrad t würde eine Abarbeitung des Jobs auf mindestens 2t+1 Prozessoren verlangen).

b) In diesem Kapitel wurde nur die Verwendung des Modells VT2 in ATTEMPTO beschrieben, wie es in der ersten Version des Systems vorgesehen ist. Denkbar ist aber auch eine Implementierung etwa des Modells VT1 oder auch der Strategie der dynamischen Diagnose. So gesehen, läßt sich ATTEMPTO auch als Hilfsmittel interpretieren, um verschiedene theoretische Modelle zu validieren, d.h. auf ihre Realisierbarkeit in Computersystemen zu prüfen.

# 7. AUSBLICK

Das Parametersystem aus Kapitel 2, das Selbstdiagnose-Modelle nach ihren wesentlichen Merkmalen strukturiert, läßt zahlreiche Modelle zu, die in dieser Arbeit noch nicht angesprochen wurden. Interessant, aber auch wegen der hohen Komplexität sehr schwierig, wäre eine Untersuchung eines Modells, das s-r-Tests mit s>1 und r>1 vorsieht (eine weitere Verallgemeinerung also von [38,73,74]). Ebenfalls, ungeklärt sind noch wichtige Eigenschaften bekannter Modelle: eine Charakterisierung der s-t-Diagnostizierbarkeit im Friedman-Modell, der sequentiellen Diagnostizierbarkeit im PMC-Modell sowie die Angabe eines (effektiven) Diagnosealgorithmus im Russell/Kime-Modell. Recht begrenzt ist die Modellbildung auch im Bereich von Ausfällen von Kommunikationsverbindungen.

Die Methode der dynamischen Diagnose läßt noch viele Probleme offen. Kann etwa gezeigt werden, daß Algorithmus A worst-case-optimal hinsichtlich der Zahl der Vergleiche ist ? Welche Möglichkeiten verbleiben der dynamischen Diagnose, wenn sie in eingeschränkter physikalischer Umgebung arbeiten soll, wenn also nicht mehr jeder Wert mit jedem anderen verglichen werden kann (etwa in einem starr vernetzten Multiprozessor-System) ? Außerdem kann dynamische Diagnose auch für Testphilosophien, die nicht auf Vergleichen aufbauen, eingesetzt werden.

Die Signaturanalyse zur Kompression von Daten bedarf noch eines geeigneteren Maßes als nur das der Fehlerwahrscheinlichkeit bei gleichwahrscheinlichen Abweichungen des Datenstroms. Insbesondere systematische Fehler, die typisch für das gerade betrachtete System sind, müssen berücksichtigt werden.

# ANHANG

Für die in Abschnitt 5.1. beschriebenen drei Algorithmen A, B und C für die dynamische Diagnose im Vergleichstestmodell VT2 werden hier Programmrealisierungen in der Programmiersprache PASCAL gegeben.
Dabei entspricht

PROGRAM DYNDIA ⟷ Algorithmus A
PROGRAM DYNDIB ⟷ Algorithmus B
PROGRAM DYNDIC ⟷ Algorithmus C .

Man beachte, daß in allen drei Programmen die maximal zulässige Wertezahl auf NMAX=17 eingerichtet ist. Bei höherer Anforderung braucht nur diese Konstantendeklaration entsprechend verändert werden.

```
PROGRAM DYNDIA;    (* DYNAMISCHE DIAGNOSE IM VERGLEICHSTESTMODELL VT2 *)
                   (* FUER N>=3, T>=1 UND T<=(N-1)/2 *)

CONST NMAX=17;
VAR N,T,R,I,J,K,L:INTEGER;
    A:ARRAY[1..NMAX] OF INTEGER;
    B:ARRAY[1..NMAX] OF -2..NMAX;


BEGIN
WRITELN('ANZAHL VON WERTEN ?');
READ(N);
WRITELN('MAX. ANZAHL VON INKORREKTEN WERTEN ?');
READ(T);
FOR I:=1 TO N DO
  BEGIN
       WRITELN('A[',I,'] EINGEBEN');
       READ(A[I]);
       B[I]:=0;
  END;
L:=N;R:=N;
WHILE (L>0) AND (T>0) DO
   BEGIN
        I:=1;J:=2;K:=0;
        WHILE (J<=N) AND (K<T) DO
           BEGIN
                IF (B[I]>=0) AND (B[J]>=0) THEN
                   BEGIN
                        IF (A[I]=A[J]) THEN
                            BEGIN L:=L-2;B[I]:=-1;B[J]:=-1;R:=I;END
                           ELSE
                            BEGIN B[I]:=J;K:=K+1;END;
                        I:=J+1;J:=I+1
                   END
                  ELSE
                   IF (B[I]<0) THEN
                       BEGIN I:=J;J:=I+1;END
                      ELSE
                       J:=J+1;
           END;
        IF ((K=T) OR (K)((3*T) DIV 4))) OR
           ((K)=((3*T) DIV 4)) AND ((((T DIV 4)*4)=T) OR
                                    (ODD(L) AND ((T MOD 4)=3)))) THEN
           BEGIN
              FOR I:=1 TO N DO
                 IF (B[I]>0) THEN
                    IF (A[R]=A[I]) THEN
                        BEGIN L:=L-2;T:=T-1;B[B[I]]:=-2;B[I]:=-1;END
                       ELSE
                        BEGIN L:=L-1;T:=T-1;B[I]:=-2;END;
           END
          ELSE
           BEGIN
              FOR I:=1 TO N DO
                IF (B[I]>0) THEN
                   IF (A[R]=A[I]) THEN
                       BEGIN L:=L-2;T:=T-1;B[B[I]]:=-2;B[I]:=-1;END
                      ELSE
                       BEGIN
                            IF (A[R]=A[B[I]]) THEN
                                BEGIN B[B[I]]:=-1;T:=T-1;END
                               ELSE
                                BEGIN B[B[I]]:=-2;T:=T-2;END;
                           L:=L-2;B[I]:=-2;
                       END;
           END;
      FOR I:=1 TO N DO
         IF (L=1) AND (B[I]=0) THEN
             BEGIN
                  IF (A[I]=A[R]) THEN B[I]:=-1 ELSE B[I]:=-2;
                  L:=L-1;
             END;

   END;

WRITELN('DER KORREKTE WERT LAUTET: ');
WRITELN(A[R]);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND KORREKT:');
FOR I:=1 TO N DO
    IF (B[I])-2) THEN WRITELN(I);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND INKORREKT:');
FOR I:=1 TO N DO
    IF (B[I]=-2) THEN WRITELN(I);
END.
```

```
PROGRAM DYNDIB;    (* DYNAMISCHE DIAGNOSE IM VERGLEICHSTESTMODELL VT2 *)
                   (* FUER N>=3, T>=1 UND T<=(N-1)/2 *)

CONST NMAX=17;
VAR N,T,R,I,K:INTEGER;
    A:ARRAY[1..NMAX] OF INTEGER;
    B:ARRAY[1..NMAX] OF -2..NMAX;


BEGIN
WRITELN('ANZAHL VON WERTEN ?');
READ(N);
WRITELN('MAX. ANZAHL VON INKORREKTEN WERTEN ?');
READ(T);



FOR I:=1 TO N DO
  BEGIN
       WRITELN('A[',I,'] EINGEBEN');
       READ(A[I]);
       B[I]:=0;
  END;
R:=N;
K:=0;
FOR I:=1 TO (N DIV 2) DO
    IF (A[2*I]=A[2*I-1]) THEN
        BEGIN B[2*I]:=-1;B[2*I-1]:=-1;R:=2*I;END
       ELSE
        BEGIN B[2*I]:=2*I-1;K:=K+1;END;
IF (K=T) THEN
    BEGIN
        FOR I:=1 TO N DO
            IF (B[I]>0) THEN
               IF (A[R]=A[I]) THEN
                   BEGIN B[B[I]]:=-2;B[I]:=-1;END
                  ELSE
                   BEGIN B[B[I]]:=-1;B[I]:=-2;END;
    END
   ELSE
    BEGIN
        FOR I:=1 TO N DO
           IF (B[I]>0) THEN
              IF (A[R]=A[I]) THEN
                  BEGIN B[B[I]]:=-2;B[I]:=-1;END
                 ELSE
                  BEGIN
                      IF (A[R]=A[B[I]]) THEN B[B[I]]:=-1 ELSE B[B[I]]:=-2;
                      B[I]:=-2;
                  END;
    END;
IF (ODD(N)) AND (R<N) THEN
    IF (A[N]=A[R]) THEN B[N]:=-1 ELSE B[N]:=-2;

WRITELN('DER KORREKTE WERT LAUTET: ');
WRITELN(A[R]);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND KORREKT:');
FOR I:=1 TO N DO
    IF (B[I]>-2) THEN WRITELN(I);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND INKORREKT:');
FOR I:=1 TO N DO
    IF (B[I]=-2) THEN WRITELN(I);
END.
```

```
PROGRAM DYNDIC;    (* DYNAMISCHE DIAGNOSE IM VERGLEICHSTESTMODELL VT2 *)
                   (* FUER N>=4 UND N/2<=T<=N-2 *)

CONST NMAX=17;
VAR N,T,R,I,J,K,L,S,U:INTEGER;
    A:ARRAY[1..NMAX] OF INTEGER;
    B:ARRAY[1..NMAX] OF -2..0;

BEGIN
WRITELN('ANZAHL VON WERTEN?');
READ(N);
WRITELN('MAX. ANZAHL VON INKORREKTEN WERTEN?');
READ(T);
FOR I:=1 TO N DO
    BEGIN
        WRITELN('A[',I,'] EINGEBEN ');
        READ(A[I]);
        B[I]:=0;
    END;
R:=N;
S:=0;
U:=0;
K:=(T DIV (N-T-1))+1;
IF ((T MOD (N-T-1))=0) THEN K:=K-1;
L:=(N DIV (K+1));

WHILE (S=0) AND (U<L) DO
    BEGIN
        U:=U+1;
        FOR I:=((U-1)*K+U) TO (U*(K+1)-1) DO
            FOR J:=(I+1) TO (U*(K+1)) DO
                IF (A[I]=A[J]) THEN
                    BEGIN
                        S:=1;
                        R:=I;
                    END;
    END;

IF ((N-L*(K+1))>=3) AND (S=0) THEN
    BEGIN
        FOR I:=(U*(K+1)+1) TO (N-1) DO
            FOR J:=(I+1) TO N DO
                IF (A[I]=A[J]) THEN R:=I;
    END;

FOR I:=1 TO N DO
    IF (A[I]=A[R]) THEN B[I]:=-1 ELSE B[I]:=-2;

WRITELN('DER KORREKTE WERT LAUTET:');
WRITELN(A[R]);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND KORREKT:');
FOR I:=1 TO N DO
    IF (B[I]>-2) THEN WRITELN(I);
WRITELN('DIE WERTE MIT DEN FOLGENDEN NUMMERN SIND INKORREKT:');
FOR I:=1 TO N DO
    IF (B[I]=-2) THEN WRITELN(I);
END.
```

# REFERENZLISTE


[ 1] J.A.Abraham,G.Metze: Roving Diagnosis for High Performance Digital Systems, in Proc. Conf. on Information Sci. and Syst., The Johns Hopkins Univ., Baltimore MD, March 78, pp.221-226.

[ 2] R.Ahlswede,I.Wegener: Suchprobleme, Teubner, Stuttgart 1979.

[ 3] F.J.Allan,T.Kameda,S.Toida: An Approach to the Diagnosability Analysis of a System, Trans. Comp., Corresp., C-24, Oct. 75, pp.1040-1042.

[ 4] E.Ammann,M.Dal Cin: Efficient Algorithms for Comparison-Based Self-Diagnosis, in M.Dal Cin,E.Dilger (Eds.): Self-Diagnosis and Fault-Tolerance, Proc., Attempto-Verlag Tübingen 1981, pp.1-18.

[ 5] E.Ammann: Vergleichstestmodelle für selbstdiagnostizierbare Systeme, Informatik-Fachberichte 54, Springer-Verlag 1982, pp.74-87.

[ 6] T.Anderson,P.A.Lee: Fault Tolerance Terminology Proposals, Proc. FTCS 12, 1982, pp.29-33.

[ 7] B.W.Arden,H.Lee: A Regular Network for Multicomputer Systems, Trans. Comp., C-31, No.1, Jan 82, pp.60-69.

[ 8] J.R.Armstrong,F.G.Gray: Fault-Diagnosis in a Boolean n-Cube-Array of Microprocessors, Trans. Comp., Corresp., C-30, No.8, August 81, pp.587-590.

[ 9] ATTEMPTO: A Fault-Tolerant Multiprocessor Working Station, Institut für Informationsverarbeitung, Universität Tübingen, Interner Bericht 7/1982/ATTEMPTO.

[10] A.Avižienis: Fault-Tolerant Systems, Trans. Comp., C-25, No.12, Dec 76, pp.1304-1312.

[11] F.Barsi,F.Grandoni,P.Maestrini: A Theory of Diagnosability of Digital Systems, Trans.Comp. C-25, No.6, June 76, pp.585-593.

[12] J.L.Bentley,H.T.Kung: A Tree Machine for Searching Problems, in Proc. IEEE Int. Conf. on Parallel Processing, 1979, pp.257-266.

[13] J.L.Bentley: A Parallel Algorithm for Constructing Minimum Spanning Trees, Journal of Algorithms, 1, 1980, pp.51-59.

[14] C.Berge: Graphs and Hypergraphs, North Holland 1973.

[15] D.K.Bhavsar,R.W.Heckelman: Self-Testing by Polynomial Division, Journal of Digital Systems, No.2/3, 1982, pp.139-160.

[16] M.L.Blount: Probabilistic Treatment of Diagnosis in Digital Systems, Proc. FTCS 7, 1977, pp.72-77.

[17] W.C.Carter: Hardware Fault Tolerance, in B.Randell,T.Anderson (Eds.): Computing Systems Reliability, London 1979, pp.211-263.

[18] K.Y.Chwa,S.L.Hakimi: Schemes for Fault-Tolerant Computing: A Comparison of Modularly Redundant and t-Diagnosable Systems, Info. and Control 49, 1981, pp.212-238.

[19] P.Ciompi,L.Simoncini: Analysis and Optimal Design of Self-Diagnosable Systems with Repair, Trans. Comp., C-28, No.5, May 79, pp.362-365.

[20] P.Ciompi,F.Grandoni,L.Simoncini: Distributed Diagnosis in Multiprocessor Systems: The MuTEAM Approach, Proc. FTCS 11, 1981, pp.25-29.

[21] J.B.Clary,R.A.Sacane: Self-Testing Computers, Computer, Oct. 79, pp.49-59.
[22] A.M.Corluhan,S.L.Hakimi: On an Algorithm for Identifying Faults in a T-Diagnosable System, Proc. 1976 Johns Hopkins Conf. Information Sci. and Systems, April 76, pp.370-375.
[23] A.T.Dahbura,G.M.Masson: A New Diagnosis Theory as the Basis of Intermittent-Fault/Transient-Upset Tolerant System Design, Proc. FTCS 12, 1982, pp.353-356.
[24] M.Dal Cin: Fehlertolerante Systeme, Teubner 1979.
[25] M.Dal Cin,E.Dilger (Eds.): Self-Diagnosis and Fault-Tolerance Proc., Attempto-Verlag Tübingen 1981.
[26] W.Diffie,M.E.Hellman: Privacy and Authentication: An Introduction to Cryptography, Proc. of the IEEE, Vol 67, No.3, March 79, pp.397-427.
[27] W.Diffie,M.E.Hellman: New Directions in Cryptography, Trans. on Info. Theory, IT-22, No.6, Nov 76, pp.644-654.
[28] D.Dolev: The Byzantine Generals Strike Again, Journal of Algorithms 3, 1982, pp.14-30.
[29] S.Even: An Algorithm for Determining whether the Connectivity of a Graph is at least k, SIAM J. of Computing, Vol 4, No.3, Sept 75, pp.393-396.
[30] T.Feng: A Survey of Interconnection Networks, Computer, Dec 1981, pp.12-27.
[31] A.D.Friedman: A New Measure of Digital System Diagnosis, Proc. FTCS 5, 1975, pp.167-170.
[32] A.D.Friedman,L.Simoncini: System-Level Fault Diagnosis, Computer, March 1980, pp.47-53.
[33] H.Fujiwara,K.Kinoshita: Testing Logic Circuits with Compressed Data, Proc. FTCS 8, June 78, pp.108-113.
[34] M.R.Garey,D.S.Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness, San Francisco 1979.
[35] S.L.Hakimi,A.T.Amin: Characterisation of Connection Assignment of Diagnosable Systems, Trans. Comp., Corresp., Jan 74, pp.86-88.
[36] F.Harary: Graph Theory, Addison-Wesley 1969.
[37] H.Hecht: Fault-Tolerant Software, Trans. on Reliability, R-28, No.3, Aug 79, pp.227-232.
[38] C.S.Holt,J.E.Smith: Diagnosis of Systems with Asymmetric Invalidation, Trans. Comp., C-30, No.9, Sept 81, pp.679-690.
[39] E.Horowitz,A.Zorat: The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI, Trans. Comp.,C-30, No.4, April 81, pp.247-253.
[40] T.Kameda,S.Toida,F.J.Allan: A Diagnosing Algorithm for Networks, Info. and Control 29, 1975, pp.141-148.
[41] A.Kavianpour,A.D.Friedman: Different Diagnostic Models for Multiprocessor Systems, in S.H.Lavington (Ed.): Information Processing 80, pp.157-162.
[42] C.R.Kime: An Abstract Model for Digital System Fault Diagnosis, Trans. Comp., C-28, No.10, Oct 79, pp.754-767.
[43] B.Könemann,J.Mucha,G.Zwiehoff: Signaturregister für selbsttestende ICs, NTG-Fachberichte, Bd.68, 1977, pp.109-112.
[44] J.G.Kuhl,S.M.Reddy: Distributed Fault-Tolerance for Large Multiprocessor Systems, Proc. of the 7th Conf. on Comp. Arch. 1980.
[45] J.G.Kuhl,S.M.Reddy: Fault-Diagnosis in Fully Distributed Systems, Proc. FTCS 11, 1981, pp.100-105.
[46] L.Lamport: The Weak Byzantine Generals Problem, Computer Science Lab., SRI International 1981.

[47] L.Lamport,R.Shostak,M.Pease: The Byzantine Generals Problem, ACM Trans. on Progr. Lang. and Syst., Vol 4, No.3, July 82, pp.382-401.
[48] D.Leisengang: Berechnung von Fehlererkennungswahrscheinlichkeiten bei Signaturregistern, Elektronische Rechenanlagen 24, 1982, H.2, pp.55-61.
[49] F.J.MacWilliams,N.J.A.Sloane: The Theory of Error-Correcting Codes, North Holland, 2.Auflage 1978.
[50] R.F.Madden: An Algorithm for System Diagnosis, Science Institute, University of Iceland.
[51] R.F.Madden: An improved bound for fault identification in t-diagnosable systems, Science Institute, University of Iceland, Oct 78, unpublished note.
[52] E.Maehle,H.Joseph: Selbstdiagnose in fehlertoleranten DIRMU-Multi-Mikroprozessor-Konfigurationen, Informatik-Fachberichte 54, Springer-Verlag 1982, pp.59-73.
[53] J.Maeng,M.Malek: A Comparison Connection Assignment for Self-Diagnosis of Multiprocessor Systems, Proc. FTCS 11, 1981.
[54] P.Maestrini,C.L.Liu: On the Sequential Diagnosability of a Class of Digital Systems, Proc. FTCS 11, 1981, pp.112-115.
[55] S.N.Maheshwari,S.L.Hakimi: On Models for Diagnosable Systems and Probabilistic Fault Diagnosis, Trans. Comp., C-25, No.3, March 76, pp.228-236.
[56] M.Malek,K.Y.Liu: Graph Theory Models in Fault Diagnosis and Fault Tolerance, J. Design Automation and Fault-Tolerant Computing, Vol 3, No.3-4, 1979, pp.155-169.
[57] S.Mallela,G.M.Masson: Diagnosable Systems for Intermittent Faults, Trans. Comp., C-27, No.6, June 78, pp. 560-566.
[58] S.Mallela,G.M.Masson: Diagnosis without Repair for Hybrid Fault Situations, Trans. Comp., C-29, No.6, June 80, pp.461-470.
[59] U.Manber: System Diagnosis with Repair, Trans. Comp., C-29, No.10, Oct 80, pp.934-937.
[60] J.A.McPherson,C.R.Kime: A Two-Level Diagnostic Model for Digital Systems, Trans. Comp., C-28, No.1, Jan 79, pp.16-27.
[61] C.Mead,L.Conway: Introduction to VLSI-Systems, Addison-Wesley 1980.
[62] G.G.L.Meyer,G.M.Masson: An Efficient Fault Diagnosis Algorithm for Symmetric Multiple Processor Architectures, Trans. Comp., Corresp., C-27, No.11, Nov 78, pp.1059-1063.
[63] G.G.L.Meyer: A Fault Diagnosis Algorithm for Asymmetric Modular Architectures, Trans. Comp., Corresp., C-30, No.1, Jan 81, pp.81-83.
[64] O.Ore: The Four Color Problem, Academic Press 1967.
[65] M.Pease,R.Shostak,L.Lamport: Reaching Agreement in the Presence of Faults, J. of the ACM, Vol 27, No.2, April 80, pp.228-234.
[66] D.K.Pradhan,J.J.Stiffler: Error-Correcting Codes and Self-Checking Circuits, Computer, March 1980, pp.27-37.
[67] D.K.Pradhan: A New Class of Error-Correcting/Detecting Codes for Fault-Tolerant Computer Applications, Trans. Comp., C-29, No.6,June 80, pp.471-481.
[68] D.K.Pradhan: Interconnection Topologies for Fault-Tolerant Parallel and Distributed Architectures, in Proc. of the 10th Int. Conf. on Parallel Processing, Aug 81, pp.238-242.
[69] F.P.Preparata,G.Metze,R.T.Chien: On the Connection Assignment Problem of Diagnosable Systems, Trans. Electron. Comp., EC-16, No.6, Dec 67, pp.848-854.

[70] B.Randell: Reliable Computing Systems, Lecture Notes in Comp. Science 60, Springer-Verlag 1978, pp.282-391.
[71] E.M.Reingold,J.Nievergelt,N.Deo: Combinatorial Algorithms Prentice Hall 1977.
[72] D.A.Rennels: Distributed Fault-Tolerant Computer Systems, Computer 13, 1980, pp.55-65.
[73] Th.Risse: Strukturelle und dynamische Eigenschaften selbsttestender Systeme, Dissertation Tübingen 1982.
[74] J.D.Russell,C.R.Kime: System Fault Diagnosis: Closure and Diagnosability with Repair, Trans. Comp., C-24, No.11, Nov 75, pp.1078-1089.
[75] J.D.Russell,C.R.Kime: System Fault Diagnosis: Masking, Exposure and Diagnosability without Repair, Trans. Comp., C-24, No.12, Dec 75, pp.1155-1161.
[76] F.Saenger,P.Bähre: Error Detection and Lokation Methods in the Real-Time and Fault-Tolerant Multi-Computer-System "RDC"-Realisation and Proposals for further Improvement, in [25], pp.201-215.
[77] F.Saheban,L.Simoncini,A.D.Friedman: Concurrent Computation and Diagnosis in Multiprocessor Systems, Proc. FTCS 9, 1979, pp.149-156.
[78] D.P.Siewiorek,R.S.Swarz: The Theory and Practice of Reliable System Design, Digital Press 1982.
[79] L.Simoncini,A.D.Friedman: Incomplete Fault Coverage in Modular Multiprocessor Systems, Proc. ACM Annual Conf. Washington, 1978, pp.210-216.
[80] L.Simoncini,F.Saheban,A.D.Friedman: Design of Self-Diagnosable Multiprocessor Systems with Concurrent Computation and Diagnosis, Trans. Comp., Corresp., C-29, No.6, June 80, pp.540-546.
[81] J.E.Smith: Universal System Diagnosis Algorithms, Trans. Comp., Corresp., C-28, No.5, May 79, pp.374-378.
[82] J.E.Smith: Measures of the Effectiveness of Fault Signature Analysis, Trans. Comp., Corresp., C-29, No.6, June 80, pp.510-514.
[83] J.Wakerly: Error Detecting Codes, Self-Checking Circuits and Applications, North Holland 1978.
[84] S.Warshall: A Theorem on Boolean Matrices, J. of the ACM, Vol 9, 1962, pp.11-12.

An dieser Stelle möchte ich Herrn Prof. Dr. M. Dal Cin für seine sehr hilfreiche und fördernde wissenschaftliche Betreuung meinen Dank aussprechen. Ebenfalls zu Dank verpflichtet bin ich Herrn Prof. Dr. W. Güttinger für die Bereitstellung der Arbeitsmittel am Institut für Informationsverarbeitung.

LEBENSLAUF

Am 21. Juni 1954 wurde ich, Eckhard Ammann, als dritter Sohn des Dipl.-Volkswirts Anton Ammann und dessen Ehefrau Gertrud, geb. Wahlenmayer, in Reutlingen geboren. Von 1961 bis 1965 besuchte ich die Grundschule in Reutlingen, ab 1965 das Johannes-Kepler-Gymnasium in Reutlingen, wo ich 1973 die Reifeprüfung ablegte. Zum Wintersemester 1973/74 immatrikulierte ich mich zum Studium der Mathematik an der Eberhard-Karls-Universität Tübingen. Dieses Studium mußte ich von März 1974 bis August 1975 unterbrechen, um im Krankenhaus Reutlingen den Zivildienst abzuleisten. Im April 1977 legte ich die Diplom-Vorprüfung, im Oktober 1980 die Diplom-Hauptprüfung in Mathematik mit Nebenfach Physik (Informationsverarbeitung) ab, das Thema meiner Diplomarbeit war "Gewichtsenumeratoren von Codes".
Von Oktober 1980 bis Februar 1981 arbeitete ich als wissenschaftliche Hilfskraft am Mathematischen Institut, seit Februar 1981 als wissenschaftlicher Angestellter über Drittmittel am Institut für Informationsverarbeitung an einem DFG-Projekt im Rahmen des Schwerpunktsprogramms "Technische Grundsatzfragen beim Einsatz von Mikroprozessoren".
Von 1981 bis 1983 fertigte ich unter der wissenschaftlichen Anleitung von Prof. Dr. M. Dal Cin am Institut für Informationsverarbeitung meine Dissertation "Modelle für die Selbstdiagnose fehlertoleranter Systeme" an.

Meine akademischen Lehrer waren vor anderen die Herren Professoren, Dozenten und Doktoren:

| | |
|---|---|
| in Mathematik: | Betsch, Hering, Heyer, Kaup, Knapp, Schönhage, Wolff. |
| in Physik: | Dal Cin, Gönnenwein, Güttinger, Lenz. |